Digital Fingerprinting of Field Programmable Gate Arrays

THESIS

James W. Crouch, Captain, USAF

AFIT/GE/ENG/08-06

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GE/ENG/08-06

Digital Fingerprinting of Field Programmable Gate Arrays

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

James W. Crouch, B.S.E.E.

Captain, USAF

March 2008

AFIT/GE/ENG/08-06

Digital Fingerprinting of Field Programmable Gate Arrays

James W. Crouch, B.S.E.E.
Captain, USAF

Approved:

/signed/	29 Feb 2008
Dr. Yong Kim (Chairman)	date
/signed/	29 Feb 2008
Lt Col James Fellows, PhD (Member)	date
/signed/	29 Feb 2008
Maj LaVern Starman, PhD (Member)	date

Abstract

Commercial off-the-shelf (COTS) component usage is becoming more prevalent in military applications due to current Department of Defense (DoD) policies. The easy accessibility of COTS will give reverse engineers a higher probability of successfully tampering, coping, or reverse engineering circuits that contain critical capabilities. To prevent this and verify the trustworthiness of hardware, circuit identification tags or serials numbers can be used. However, these values can be easily obtained and forged. To protect critical DoD technologies from possible exploitation, there is an urgent need for a reliable method to confirm a circuit's identity using a set of unique unforgettable metrics.

This research proposes the concept of creating a circuit identifier, or digital fingerprint, for application specific integrated circuits (ASIC) and field programmable gate arrays (FPGAs). The digital fingerprint would be a function of the natural variations in the semiconductor manufacturing process and the functionality of the circuit allowing the creation of a unique identifier for a specific chip that can not be duplicated or forged.

The proposed digital fingerprint allows the use of any arbitrary node or set of nodes internal to the circuit and the circuit outputs as monitoring locations. Changes in the signal on a selected node or output can be quantified digitally over a period of time or at a specific instance of time. Two monitoring methods are proposed, one using cumulative observation of the nodes and the other samples the nodes based on a signal transition.

Testing of the two monitoring methods was performed on a small sample of twenty Xilinx® Virtex-II Pro FPGAs. Both methods successfully created unique identifiers for each FPGA.

Acknowledgements

I would like to thank my parents for being great sounding boards for my ideas and always standing by me.

I express my astonishment and appreciation to my faculty advisor, Dr. Yong Kim for dealing with me on a daily basis.

My thanks to Lt Col James Fellows and Maj LaVern Starman for their review of, and recommendations for improvement to, this long, painful document.

Finally, I give my heartfelt appreciation to the rest of the VLSI group of 08M. We went through the good and bad times, but we always supported one another.

To paraphrase Leonardo da Vinci: No document is ever finished, just abandoned.

James W. Crouch

Table of Contents

Page

List of Figures

List of Tables

Table Page

List of Abbreviations

Abbreviation		Page
ASIC	Application Specific Integrated Circuit	1
DoD	Department of Defense	1
COTS	Commercial off-the-shelf	1
FPGA	Field Programmable Gate Array	1
VHSIC	Very High Speed Integrated Circuit	3
VHDL	VHSIC Hardware Description Language	3
IC	Integrated Circuit	5
I/O	Input/Output	5
CLB	Configurable Logic Block	5
LUT	Look-Up Table	6
DFF	D Flip-Flop	6
MUX	Multiplexer	6
SRAM	Static Random Access Memory	6
ATE	Automatic Test Equipment	9
FIB	Focused Ion Beam	10
SEM	Scanning Electron Microscope	10
SPA	Simple Power Analysis	12
DPA	Differential Power Analysis	12
JTAG	Joint Test Action Group	13
PROM	Programmable Read-Only Memory	14
PUF	Physical Uncloneable Function	14
RO PUF	Ring Oscillator PUF	15
LSFR	Linear Feedback Shift Register	21
MSB	Most Significant Bit	21
LSB	Least Significant Bit	21

Abbreviation		Page
NCS	Nodal Cumulative Sampling	24
XUP	Xilinx® University Program	27

Digital Fingerprinting of Field Programmable Gate Arrays

I. Introduction

War never changes. One of war's major facets is the clash between the technology of opposed civilizations. From copper sword bearing tribes that were destroyed by those equipped with iron to the French armies that suffered greatly under English longbowmen, technological advancement over ones enemies is a critical edge. The United States military has excelled in conventional warfare because of the continual push to advance its technological capabilities. Foreign governments attempt to counter this by either purchasing critical components [13] or salvaging them. Regardless of how they are acquired, the critical components that provide the U.S. military with its technological edge will find their way into enemy hands.

Typically, only a sample or two of an Application Specific Integrated Circuit (ASIC) falls into enemy hands and the reverse engineering process requires multiple samples as the circuit is destroyed in the process. This is changing with the Department of Defense (DoD) pushing more military systems to use commercial off-the-shelf (COTS) parts [8]. This change is due to the need of deploying new capabilities to the warfighter. The preference is to give warfighter the 80% solution now, rather than the 100% solution in five to ten years as has been done in previous years. Utilizing COTS components and integrating them rather than doing a complete research and development effort from scratch drastically reduces the time required to deliver new capabilities. Additionally, the use of COTS provides additional program savings due to ease of availability and inexpensiveness. This switch to COTS also involves a switch to commercial ASICs and Field Programmable Gate Arrays (FPGAs). Unfortunately the same reasons that make COTS ASICs and FPGAs advantageous are also

liabilities, namely being inexpensive and their easy of availability on the commercial market.

This makes reverse engineering and tampering simpler as multiple samples can be easily purchased with no governmental oversight. Once the functionality is reverse engineered, design of a functional equivalent circuit is trivial and the technical advantage is lost. Additionally, hidden lethal functionality could be added and the now tainted ASICs could be returned to government control, either via normal acquisition channels or through the return of equipment loaned to U.S. allies.

This potential threat has become more prevalent due to the increased use of FPGAs instead of traditional ASICs. Program managers have increased FPGA usage due to their lower life-cycle cost. From a programmatic view, FPGAs provide a lower cost solution in three areas due to their reprogrammability. First, the design-implement-test cycle is far faster because new functionality does not have to be shipped out to a fabrication center and can be done on-chip. Second, out-year fund savings occur as systems that use FPGAs can be easily updated with new functionality, allowing new capabilities to be used by the warfighter much faster than the acquisition system would normally allow. Third, ASICs typically undergo a redesign every time there is an improvement in fabrication technology costing time and money. FPGAs are downwards compatible, meaning that old configuration programs are just recompiled for new FPGA designs.

1.1 Problem Statement

Due to the increasing use of non-military proprietary hardware, adversaries may have increased their chances of learning how COTS-based military hardware functions through reverse engineering the COTS components. A method is needed to uniquely identify each piece of hardware and/or generate an unique signature that can not be duplicated or fabricated by adversaries. Such a signature would allow easy verification of loaned hardware to prevent modified or incorrect circuits with hidden functionality from entering the acquisition system. Additionally, it could provide

hardware unique encryption and decryption keys. The nature of this signature will require the hardware functionality be tied to the physical silicon on which it resides. The first step in signature creation is identifying a methodology for differentiating multiple functionally and structurally identical circuits from the same vendor.

1.2 Goals

The goals of this research are:

1. Assess circuit signal attributes that can potentially serve as part of an identifier
2. Identify the most likely attribute or attributes that can be used on both ASIC and FPGA circuits
3. Validate the proposed attributes in an FPGA architecture

1.3 Scope and Assumptions

This research uses the Xilinx Virtex-II Pro architecture to show the digital fingerprint concept due to the time and cost constraints on fabricating an ASIC test circuit. Implementation can be performed on any FPGA, but is being done on the Virtex-II Pro architecture due to easy access to multiple samples.

1.4 Methodology

Multiple physical signal attributes are examined as potential candidates for use in creation of a digital fingerprint. One or more will be selected and be used as the basis of a unique identifier. Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) code is used to implement and show this on a reconfigurable FPGA.

1.5 Materials and Equipment

The following software and hardware is used in this research: Mentor Graphics ModelSim PE Student Edition 6.3c, MathWorks `Matlab`® 6.5, Xilinx ISE 9.2i and Virtex-II Pro XC2VP30 FPGA, and Agilent 16902 Logic Analyzer.

1.6 Overview

This research is organized in the following manner. Chapter II contains an introduction to the architecture of the Virtex-II Pro, tamper methods for reverse engineering and altering ASIC and FPGAs, and other research that attempts to create unique identifiers. Chapter III describes the digital fingerprint implementation that is examined in this research. Chapter IV provides the test circuits for, and the results gathered from, the digital fingerprint implementation in Chapter III with Chapter V summarizing and concluding this research.

II. Background

This chapter provides an overview of how a FPGA is designed structurally and operates as the Digital Fingerprint concept described in Chapter III is implemented on a commercial FPGA. Some of the tamper methods that can be used to alter or allow the theft of the functionality in an integrated circuit (IC) are examined. Finally, a summary of other efforts to create a unique identifier for ICs is provided.

2.1 Field Programmable Gate Array

A FPGA is a special type of ASIC that has an architecture which allows the implementation virtually any functionality. ASIC fabrication has a large development cost and any change to the underlying functionality or in the fabrication feature size necessitates a redesign and new fabrication run. As a result, the FPGA has become popular in the commercial market, and by extension in DoD acquisition programs, due to the easy of implementing new functionality. Configuring a FPGA involves a design engineer writing a hardware description in a pseudo-computer language, VHDL or Verilog, and downloading it to the FPGA to run. Updates are as simple as adding a few new lines of code, running workbench tests on a single chip, and uploading the new code.

2.1.1 Architecture. The FPGA is a special type of ASIC that is designed to have reconfigurable functionality based on a string of input bits. The basic structure of any FPGA can be seen in Figure 2.1. An FPGA consists of four main components: the input/output (I/O) pins, which allow communication between the internals of the FPGA and the outside world; the configurable logic blocks (CLBs) where the actual functionality of the circuit loaded into the FPGA resides; the routing and routing switches which connect up the CLBs and the I/O pins. The amount of space used by all the CLBs on a chip is small, relative to the connections between CLBs which take up the bulk of the silicon. Due to the reconfigurable nature of the FPGA, these connections need to be large and versatile to allow a wide variety of routings. Figure 2.2 shows the internals of a basic CLB. Inside every CLB is three components,

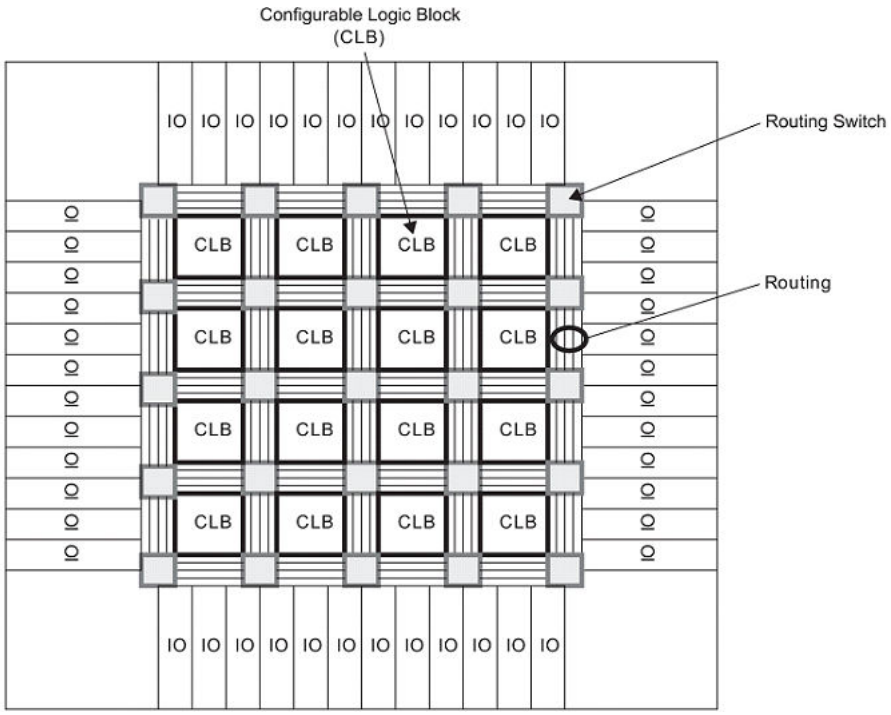


Figure 2.1: Basic structure of an FPGA, consisting of I/O pins, CLBs, routing wires, routing switches [19].

a Look-Up Table (LUT), D Flip-Flop (DFF), and multiplexer (MUX). The LUT is a four input truth table that can implement any four variable Boolean equation and provide a single output. The DFF is a simple storage element used for sequential circuits whose output takes on the input value upon the reading of a clock edge. The multiplexer allows the setting of the CLB to either perform combinational logic by passing the LUT output directly or sequential logic by passing the DFF output. In reality the CLB is much more complex as Figure 2.3 shows one of four slices in a CLB.

2.1.2 Configuration. Configuration of a FPGA is done using a data stream known as the bitstream. The bitstream contains all the configuration information for the LUTs, multiplexers, DFFs, and CLB interconnects and is both manufacturer and chip type dependent. Commonly, the bitstream is stored in static random access memory (SRAM) cells on the FPGA. SRAM, maintains its contents as long as it is powered, but being a volatile memory type requires a reload of the bitstream

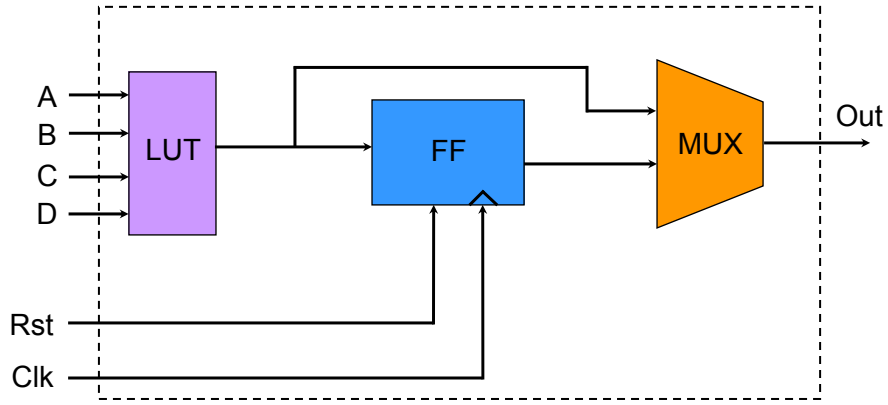


Figure 2.2: The structure of a basic CLB.

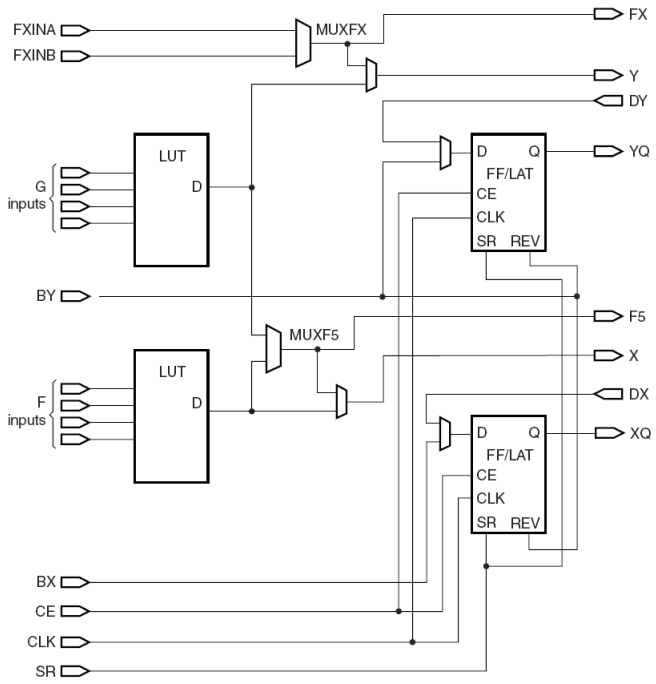


Figure 2.3: The internal structure of a Xilinx® Virtex-II Pro CLB slice [22].

anytime there is a power loss. Some FPGAs use non-volatile memory such as Flash or antifuses. These are mainly used on FGPAs that require extra security as volatile memory FPGAs require off chip bitstream storage and provide an additional attack vector for the reverse engineering process.

It is worth noting that a change in the functionality that is to be programmed into the FPGA will result not only a different bitstream but a different part of the FPGA being configured to implement that functionality. As designers typically do not care how the internal configuration of the FPGA is done, this aspect of FPGA utilization is typically overlooked but will play a critical part in the implementation of the digital fingerprint on an FPGA.

2.2 Tamper Methods

There are a number of different ways of discovering and altering the functionality of a circuit. Collectively, these are called tamper methods and can apply to either ASICs or FPGAs.

2.2.1 Brute Force Attacks. Brute force attacks interact with an IC through its I/O pins and are performed either while the IC is in natural environment or stand-alone.

2.2.1.1 Black Box Attacks. Traditionally, the black box attack has been the first and simplest attack to perform on an IC in an attempt to reverse engineer it. Chips are examined individually, all possible input combinations applied and the outputs recorded. Using a large truth table, data analysis algorithms, or in some cases visual inspection, the underlying Boolean equations that define the IC's logic can be recreated [21]. This type of attack only works well on circuits with well defined inputs and outputs, combinational circuits, and latches [9]. Synchronous circuits, like state machines, increase the complexity of reverse engineering as an

incorrect input at the wrong time may cause the circuit to miss a state transition or reset, resulting in an incomplete understanding of the circuit.

With FPGAs, the challenges are greater, due to the number of pins that can be assigned as either inputs or outputs, and in some cases both. There are 2^n input combinations on any IC, where n is the number of input pins [6]. The basic Xilinx® Virtex-II Pro XC2VP2 introduced in 2002, has 204 pins that can be designated as either input, output, or bi-directional [23]. If we make the simplifying assumptions that the circuit is purely combinational, and half the pins function as inputs so $n = 102$, then there are:

$$2^{102} = 5,070,602,400,912,917,605,986,812,821,504$$

or $\approx 5 * 10^{30}$ combinations to apply to the circuit.

If the reverse engineer, with no prior knowledge of the FPGA, utilizing a state-of-the-art automatic test equipment (ATE) running at 1 GHz (roughly $1.5M), picks the correct pins to apply the input on the first try, it would take:

$$5 * 10^{30} inputs * \frac{sec}{10^9 input} * \frac{min}{60 sec} * \frac{hr}{60 min} * \frac{day}{24 hr} * \frac{year}{365 hr} \approx 1.6 * 10^{14} years$$

to examine the entire space of input combinations. This is long after the Earth has become nothing more than a burnt cinder around a dead sun.

2.2.1.2 Passive Attacks. The next step up from the black box attack is the passive attack. This attack is conceptually very simple. ICs are examined in their native environment, i.e. while they are being used in an actual circuit. Input and output lines are monitored, typically using either an oscilloscope or logic analyzer, and data is recorded giving a good picture of the chips functionality. This attack takes far less time than the black box attack as an exhaustive search of all possible input combinations is not necessary. However, if the chip being examined is synchronous in nature, it is possible that not all states are seen leaving out vital functionality.

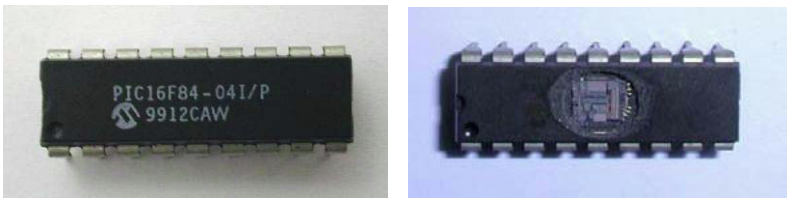


Figure 2.4: PIC16F84 microcontroller packaged and depackaged [17].

Typically, this attack is used as an initial probe to provide focus for a follow-on black box attack for state machines. For FPGAs, the passive attack greatly simplifies the black box attack search space as the input and output pin are readily identified.

2.2.2 Hardware Attacks. Hardware attacks focus on the physical silicon of the chip. Typically, this involves depackaging the IC, as shown in Figure 2.4, although for some attacks this is unnecessary. This allows direct interaction at the transistor level without having to go through the I/O pins, allowing a better functional understanding. These attacks carry the risk of destroying the IC as they are directly interacting with the silicon. Consequently, multiple samples of the IC being attacked are required. In the case of commercial chips, this is a fairly simple task, resulting in a higher likelihood of successful reverse engineering.

2.2.2.1 Mechanical Probes. Mechanical probing is the traditional method of reverse engineering post-depackaging of the IC. Small metal probes are used to measure voltage and current at the transistor level, providing a finely detailed look at the circuit functionality. As physical contact is required to make these measurements, there exists the chance that the probe contact will destroy some of the transistor or wires.

2.2.2.2 Focused Ion Beam. The focused ion beam (FIB) is a semiconductor fabrication device used as a micro-machining tool. It has three modes of operation. First the FIB can image in a similar method to the scanning electron microscope (SEM) but using gallium ions instead of electrons. Unlike the SEM, it has a destructive effect as the gallium ions are implanted into the sample surface changing

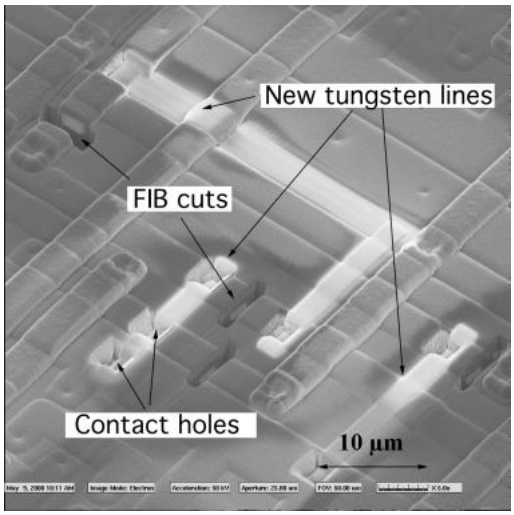
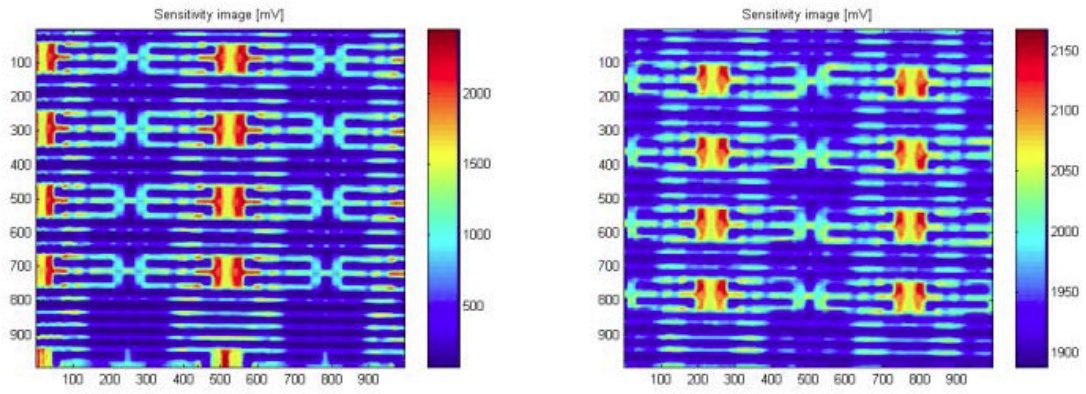


Figure 2.5: FIB created image of a circuit altered by the FIB's micro-machining capabilities [15].

the nature of the underlying surface through introduction of crystalline defects or additional electrons. The second and third modes are its micro-machining capabilities, allowing milling or metal deposition, typically tungsten [10]. Figure 2.5 shows an FIB image of a chip that has had both milling and metal deposition performed on it.

2.2.2.3 Optical. Optical tamper methods are semi-invasive as they require the depackaging of the IC but do not have any physical interaction with it. Instead, they rely on the interaction of photons with semiconductor devices. The photons must have an energy higher than the bandgap of the material being examined, 1.12 eV for silicon. Lasers are often used as they can be focused on a small area. These attacks take two forms: optical probe and optical attack. Optical probing focuses on circuit examination by looking at transistor states [16]. Figure 2.6 shows an SRAM cell under optical probe and was created by measuring the photocurrent induced by the laser scanning the SRAM. Color indicates the measured photocurrent,. The optical attack works by activating transistors in the same way that exposure to gamma radiation would, creating free electrons by exciting them to the conduction band causing a bit flip. The effect on the IC, e.g. broad area ionization, is not as



(a) (b)

Figure 2.6: Examination of SRAM cells. (a) Shows the cells unpowered. (b) 16 SRAM cells in a 4x4 grid loaded with values. The first row is read 1100 [16].

great as gamma irradiation due to the lower energy laser generated photons and the small area they are applied.

2.2.3 Side Channel Attacks. Rather than using brute force to physically probe the internals of an IC, areas that leak unintended information known as side channels are used. Side channel attacks use this secondary information to create a picture of IC functionality. They include power consumption and timing analysis.

2.2.3.1 Power Consumption. Power consumption attacks originated for, and still mainly focus on, breaking cryptography schemes. For reverse engineering, the concept is that through an examination of the power utilization of an IC, the underlying circuit functionality can be found. There are two types of power consumption attacks: Simple Power Analysis (SPA) and Differential Power Analysis (DPA) [14]. SPA consists of creating power consumption vs time graphs, captured at a high sample rate, of the power or ground input. Analysis of the data can show how the chip operates on the instruction level [3]. This attack can be done using a decent quality digital oscilloscope.

DPA is similar to SPA in the use of power consumption vs time graphs. Additionally, DPA uses statistical analysis and error-correction statistical methods [3].

This two-pronged approach gives an attacker insight into the data values that are being manipulated on a chip. It is possible to then correlate this collected data to known functions in order to see exactly what is happening. This attack, due to the extra analysis, takes longer than SPA, but provides a better picture of the internal functionality.

2.2.3.2 Timing Analysis. With brute force attacks, synchronous circuits add additional complexity in the reverse engineering process due to the timing constraints that are introduced. For circuit designers, synchronous circuits are much easier to design as the clock functions as a control signal allowing data to be sent and received at specific times. However, the predictability of the clock signal and its permeating nature throughout the circuit can be used as an attack vector. Timing attacks focus on taking the circuit outside normal parameters by modifying the speed of the clock, either speeding up or slowing it down. The change in clock speed can cause a synchronous circuit to miss a clock edge and enter into an incorrect state or miss a state transition entirely, while stopping it will allow an attacker to see what is happening internally at any clock cycle.

2.2.4 FPGA Specific Attacks. The previously mentioned attacks work on both ASICs and FPGAs. As FPGAs have the ability to be reconfigured, there are additional attacks that can be used by the reverse engineer that focus on the configuration bitstream.

2.2.4.1 Readback Attacks. A common feature of FPGAs is the ability to read the configuration data out. This feature is used to troubleshoot programming that has been loaded onto the FPGA. In this attack, the configuration data is read from the FPGA via the programming or Joint Test Action Group (JTAG) port. Normally, this functionality is disabled through the setting of a security bit. The security bit, while fairly well protected, can be neutralized by using a fault generating attack, like the optical attack.

2.2.4.2 Cloning Attacks. In SRAM and other volatile memory based FPGAs, the bitstream is stored off-chip in non-volatile memory, usually programmable read-only memory (PROM), and is loaded in on power-up. This provides an attacker easy access to the bitstream by just monitoring the bus lines from the PROM to the FPGA. The bitstream then can be easily loaded into another FGPA creating an exact clone. This attack is extremely useful as it allows any number of functional copies to be made and be reverse engineered in parallel.

2.3 Physically Uncloneable Functions

A review of current literature shows that most efforts in the creation of unique identifiers focus on the watermarking of intellectual property, mainly in terms of functionality implemented on FPGAs. One group has looked at the creation of a unique identifier for both ASIC and FPGAs.

2.3.1 Background. In 2002, MIT published a paper, [11], on what they called silicon physical random functions. These functions, later renamed physical uncloneable functions (PUFs), are stand-alone structures specifically designed to create an ID via variations in internal delay of the PUF function due to manufacturing process variations. As of June 2007, MIT has proposed and tested two PUF circuits based around this concept, an arbiter and a ring oscillator [18]. Both of these structures generate their IDs based on the relative difference of the transient response of the wire delay after an input is applied.

2.3.2 Arbiter PUF. The arbiter PUF uses a single signal traveling down two wires of identical length. The wires pass through multiple switching boxes, consisting of two MUXes each, to a DFF at the end. Control of the switching box array is done via the PUF input bits, as shown in Figure 2.7. A value of X=0 has the signals pass though while X=1 causes the signals to switch lines. As the signals propagate through the circuit, variations in the wire will cause changes in the signals' delay. The DFF outputs a 0 or 1 depending on which signal arrives first.

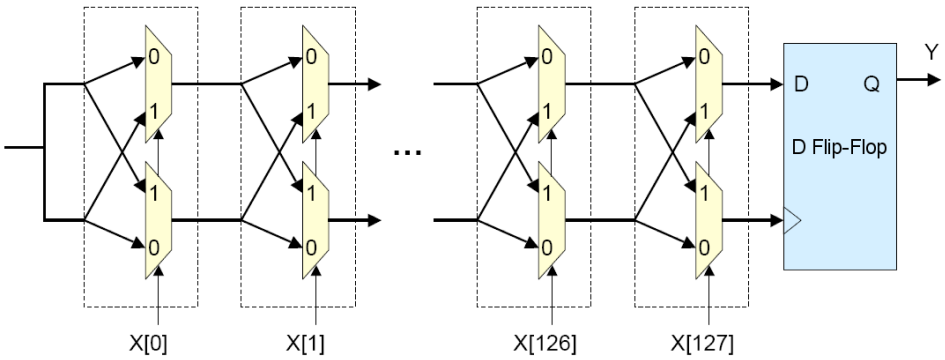


Figure 2.7: MIT's Arbiter PUF. Signals can switch lines depending on MUX input bits. The output Y reports which path was faster [18].

To generate an ID that is unique over multiple ICs, one of two methods can be used. First is to apply n different input vectors to a single arbiter PUF to get an output of length n. Second is to replicate the PUF n times and apply one input to all PUFs simultaneously giving an n-bit output.

2.3.3 Ring Oscillator PUF. The ring oscillator PUF, or RO PUF, uses variations in frequency of many identically laid-out ring oscillators to generate a circuit ID. Manufacturing variations will cause changes in wire and gate delay, altering the frequency of each oscillator and the number of pulses generated over a time interval. The oscillators are selected in pairs, via MUX, and the number of pulses over an arbitrary length of time counted. The output is based on which oscillator had the larger number of pulses, as shown in Figure 2.8.

Generating a unique ID is harder for RO PUFs than arbiter PUFs. Normally for N oscillators there are $N(N-1)/2$ possible combinations, giving that many identification bits. Due to correlation, if the number of pulses of oscillator A is greater the number of oscillator B and B > C, then the number of pulses of A > C. The actual number of possible combinations now becomes $log_2(N!)$. Therefore, 128 oscillators produce a maximum of 716 bits and 1024 oscillators produce 8769 bits and so on. Alternatively, each oscillator pair could only be used once so 128 pairs of oscillators, 256 total, would give a 128-bit ID.

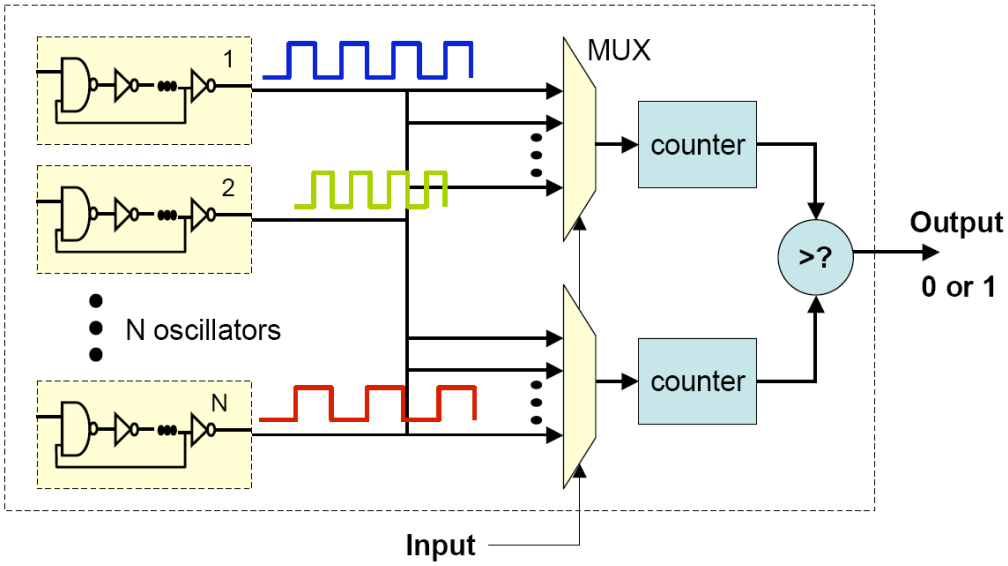


Figure 2.8: MIT's Ring oscillator PUF [18].

2.3.4 Examination of the RO PUF. In Suh et al [18], results of an RO PUF implemented across 15 Virtex-4 LX25 FPGAs are reported. A 128-bit FPGA signature is generated using 1024 ring oscillators, each consisting of five inverters and one AND gate. As only 256 oscillators are needed to generate a 128-bit output, using each RO once, the additional ring oscillators are used to ensure a large enough frequency difference is present between pairs to generate reliable outputs. This results in only a quarter of the total number of ring oscillators being used to generate an identification signature. By using each oscillator only once, only 1.5% of the total number of potential identification bits are used.

The data shows an average variation between FPGAs of 46.15% when comparing output bits across all 15 FPGAs. However, this data only uses 105 out of the 128-bits generated. No mention is made of the missing 23 bits. If there was no change in their output they would have been counted. One can deduce that the outputs were not stable and provided fluctuating values on the counters over multiple runs. For the 105 bits that were used, no mention is made of whether it is always the same bits or different groupings of 105 out of the 128 bits. Also not mentioned is the effect of LUT simplification on the RO PUF. The Xilinx® ISE reduces combinational logic

structures down to Boolean equations for instantiation as LUTs. The ring oscillators used in [18] consist of five inverters and one AND gate, which reduce to a single LUT. Without more information it is hard to elaborate on, or evaluate, the performance of the MIT PUF design. Regardless, it can be concluded based on the designs given that PUFs rely on a finalized stable output values to characterize the circuit.

III. The FPGA Digital Fingerprint

The creation of a unique identifier for a chip has traditionally been done by industry via use of a serial number, either imprinted on the chip package or applying it to the chip after fabrication, like Intel's CPU ID on the Pentium III. Neither of these IDs is secure, as the first can be negated by producing either a counterfeit package or replacing the chip inside the a legitimate package with another. The second ID, being applied after fabrication is still susceptible to either change or copying by a third party.

3.1 The Digital Fingerprint - Overview

The digital fingerprint creates an unique identifier based on the characterization of signals that are unique to a particular physical implementation of a functionality. Ideally, all chips carrying the same circuit design and fabricated in the same way will be exactly identical. This is not true. The semiconductor fabrication process is not perfect and minor variations exist in each chip. These variations can, for example, happen in the doping profile, mask alignment, metal deposition, oxide growth, or transistor gate width and length and have been proven to have a statistically significant effect on some circuit attributes [4, 5]. The effect that these variations have can be seen in Intel's processor lines. In examining their Core 2 Duo with 1333 MHz front-side bus, the clock speeds for the E6850, E6750, and E6550 CPUs go from 3 to 2.66 to 2.33 GHz respectively [7]. Fabrication facilities are not cheap, costing in the billions of dollars, and run twenty-four hours a day. No profitable corporation would purposely fabricate three different processors with identical specifications except for small differences in clock speed as it would impact long term profitability.

The waveforms that occur on a signal line inside a circuit are dependent on both the circuit functionality that drives the signal line and the variations of the physical implementation of that functionality. A fingerprint that examines the signal characteristics at a specific node can create an identifier that will be unique to a chip due to these two dependencies.

3.2 Circuit Attributes

It is commonly believed by people today that they live in a digital world with all the high-tech devices like computers, cell phones, and high-definition television operating by 1's and 0's. The truth is that the world is actually analog and that the devices considered digital are, in reality, dealing with analog signals that are interpreted as digital values. These analog signals allow us to use a number of circuit attributes to create a digital fingerprint. The attributes are based on the design of the functional logic that is being implemented and the variations in physical implementation and have an effect both on a signal and on how it is interpreted by the surrounding logic.

3.2.1 Voltage. The voltage on a signal line generated by a circuit determines whether other circuit components will interpret a logic 0 or logic 1.

3.2.2 Noise Margin. The noise margin is the range that an input voltage to a circuit component can be so that it is interpreted as a logical 0 or 1. For a signal that can swing between 0 V and 5 V, the noise margin may be 0.5 V for a logic 0 and 4.5 V for a logic 1.

3.2.3 Current. Voltages move on a signal line due to the current also flowing. Changes in the current can result in some voltage levels not reaching other circuit components.

3.2.4 Setup/Hold Time. In combinational circuits, changes at the inputs can happen at any rate with no adverse affects on the output value other than glitches. Flip-flops and latches, due to the cross-coupled feedback loops in the NAND gates of the basic cell, as shown in Figure 3.1, have input constraints. Setup time is how long the D input must remain constant until the clock input reaches 50% max, assuming a rising edge triggered D flip-flop. Hold time is the time that the D input must be held constant after the 50% point of the clock input to ensure that metastability doesn't

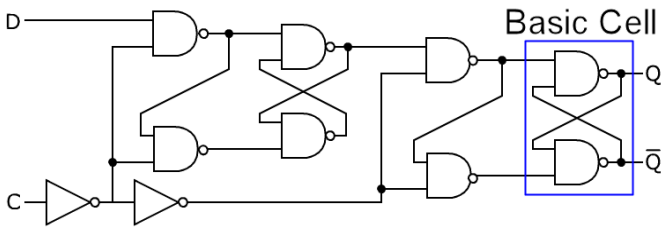


Figure 3.1: A Master-Slave D Flip-Flop with highlighted basic cell.

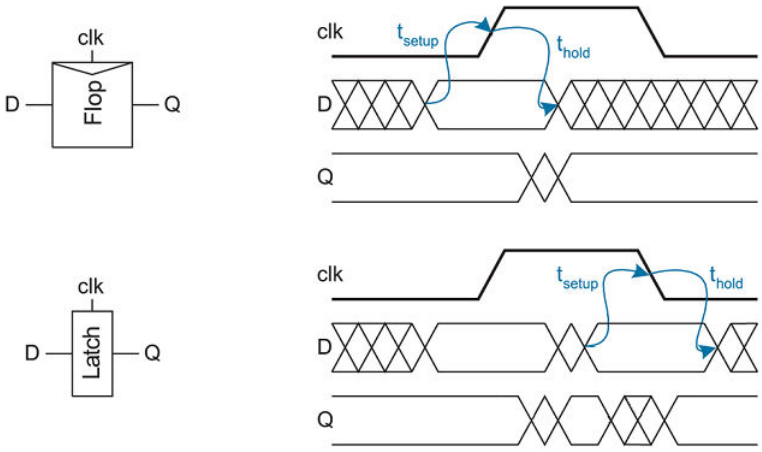


Figure 3.2: Setup and hold times for the D flip-flop and latch [19].

happen. For a latch, setup and hold times must be observed around the falling clock edge. Simply, there is a time t that the input must hold constant around a clock edge for proper functionality. Figure 3.2 shows this pictorially. In the case where either of these times is violated, the output response is as shown in Figure 3.3.

3.2.5 Delay. There are several different delays, which are described below.

3.2.5.1 Contamination Delay. Contamination delay is the minimum time required for a logic gate or combinational circuit input, measured at 50% of its final value, to effect it's output, again measured at 50% of its final value [19].

3.2.5.2 Propagation Delay. Similar to contamination delay, propagation delay is the maximum time required for an input measured at 50% of its final value to cause the output to reach 50% of its final value [19]. Between the contami-

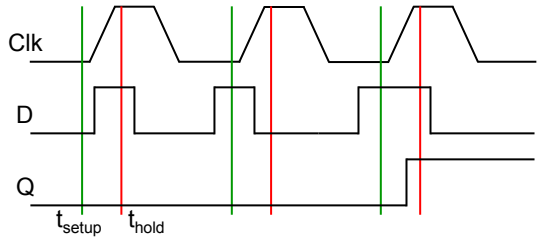


Figure 3.3: Setup and hold time violations for a D Flip-Flop. The first and second pulses violate a flip-flop's setup and hold time respectively, resulting in no change in the output. The third pulse is wide enough that it meets both time requirements and results in a 1 on the flip-flop output.

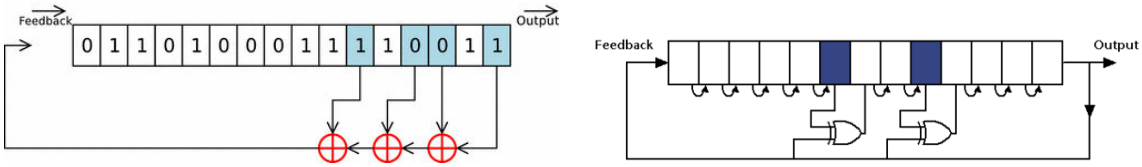
nation and propagation delays, the output of the combinational circuit may begin to glitch, or show multiple transitions between logic 0 and logic 1, as the logic gates and paths through the combination logic to the output will have non-uniform contamination and propagation delays. It can be argued that propagation and contamination delay are analog in nature. However, if you look at their effect on a output, they can be considered digital.

3.2.5.3 Critical Path Delay. Critical paths in a logic circuit, whether combinational or sequential, are the slowest and limit the speed of the circuit. The critical path delay is an extension of the propagation delay and is the longest amount of time in a circuit for a change in the input to change the output.

3.3 Stand-alone Structure Digital Fingerprint

This section looks at an attempt to create a digital fingerprint via a stand-alone structure using an asynchronous linear feedback shift register (LSFR).

3.3.1 LFSR Architecture. The LFSR is a standard DFF based shift register with two additional features: a feedback loop from the most significant bit (MSB) output to the least significant bit (LSB) input, and XOR gates, called taps, causing the LSB to be a linear function of the previous state [1, 6].



(a) (b)

Figure 3.4: LFSR in (a) external configuration [20] and (b) internal configuration [20]. The squares highlight which registers are connected to the XOR taps.

3.3.2 Types of LFSR. There are two ways of positioning the taps, creating two configurations or types of LFSR, formally known as Fibonacci and Galois, but more commonly referred to as external and internal respectively. In external LFSRs, taps are placed in the feedback loop, creating a mod-2 sum of all the taps to appear on the input. For internal LFSRs, the taps are placed between the flip-flops, as shown in Figure 3.4. Both architectures have low gate delays, however the parallel nature of XOR positioning in the internal design allows smaller delays than the external [12].

3.3.3 LFSR Operation. The output of the LFSR is controlled by three parameters: clock, tap positions, and the initial value that is loaded into the LFSR or seed. Depending on where the taps are placed, the LFSR will have two or more state spaces that it can potentially cycle through. Which state space used is dependent on the seed value. For every n-bit LFSR there is a set of tap positions that will give only two state spaces. One space has a length of $2^n - 1$ states, known as the maximal length, and the other consisting of the all 0 state. Typically used is the tap configuration to produce the maximal length state space as the MSB output acts as pseudo-random number generator. The period of the pseudo-random numbers, i.e. until they begin to repeat, is equal to the size of the state space.

3.3.4 Asynchronous LFSR. LSFRs use DFFs for bit storage which only update their value on a clock edge. Latches operate nearly the same, but have an additional quality, transparency. Transparency causes the latch to update its output while the clock is high. Figure 3.5 shows a timing diagram with the output of a

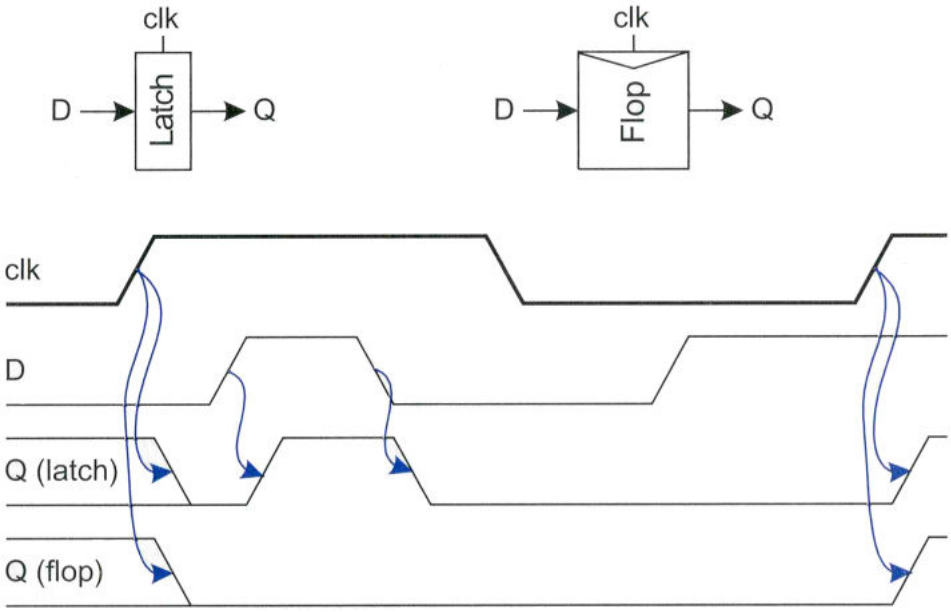


Figure 3.5: DFF and Latch timing diagram for output Q in response to input D w/clock [19]. This lines indicate the relationship between the signal transitions. Note the latch shows transparency by changing its output in response to D while clk is high.

flip-flop and a latch. Common practice is to avoid transparency as a system with it will behave unpredictably. However, with the transparency of the latch the states will continue to change during the half clock cycle in which the clock is high. When the clock goes low, the setup and hold times of latches come into play and any transition on the latch input that violates these will not be seen. These times along with the latches' internal delay and the delays of the taps and wires, will determine the final state that the modified LFSR is in. As these delays vary across FPGAs, the final state should be different for every FPGA. The digital fingerprint for the circuit, then, is the value stored in the LFSR after a clock pulse. As the clock pulse that triggers the LFSR will also have some variation, this will add an additional factor to the creation of a digital fingerprint.

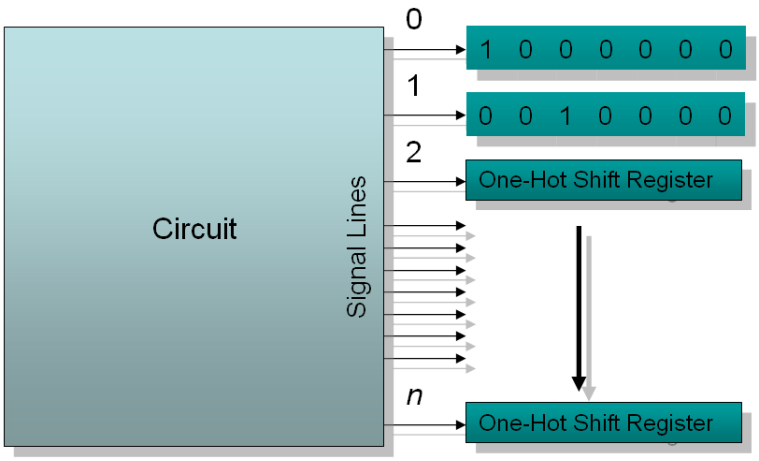


Figure 3.6: Block structure of NCS fingerprint method.

3.4 Transition Based Fingerprints

Instead of using a stand-alone structure to create a digital fingerprint, we can use the transitions between '0' and '1'.

3.4.1 Nodal Cumulative Sampling. The nodal cumulative sampling (NCS) method involves the summation of transitions, either $0 \rightarrow 1$ or $1 \rightarrow 0$, over multiple signal lines to create a digital fingerprint. These lines are from various places in a circuit and are connected to the clock input of one-hot-encoded shift register, as shown in Figure 3.6. The register has a '1' value on the LSB and as transitions are detected on a signal line, '0's are shifted in causing the '1' to shift towards the MSB. The bit that the '1' ends on after all transitions are counted is the fingerprint value for that line. Performing this process over multiple lines results in a base-n digital fingerprint, where n is the maximum number of transitions seen for the signal set. Variations in the signals and the setup/hold times of the one-hot shift register will result different fingerprint values across multiple circuit implementations. As an example, Figure 3.7 shows the basic concept by performing the summation of the $0 \rightarrow 1$ transitions for two signal lines. The fast $0 \rightarrow 1$ transition in the bottom signal may or may not be captured by the shift register depending on its setup/hold times resulting in two unique identifiers.

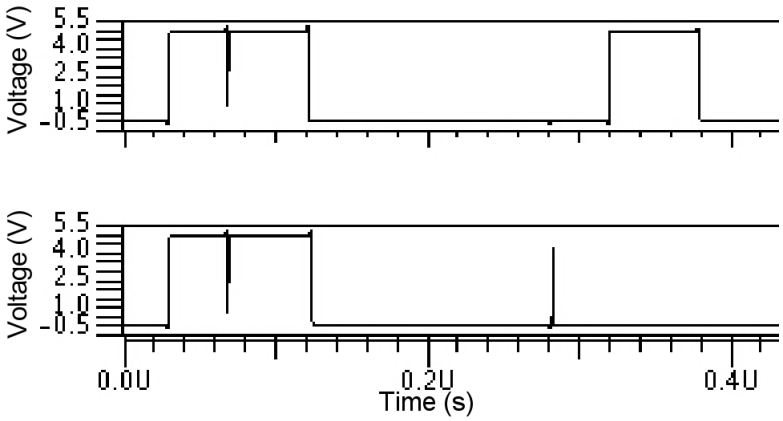


Figure 3.7: Arbitrary signals with transitions.

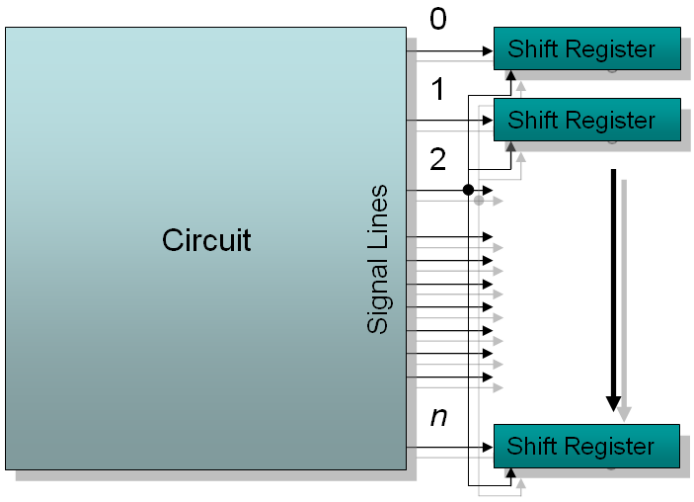
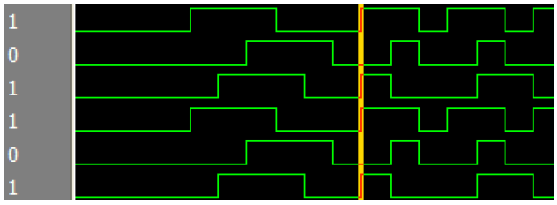
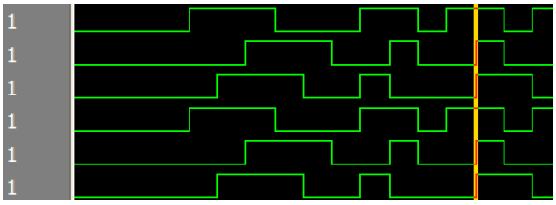


Figure 3.8: Block structure of the transitional sampling fingerprint method w/signal 2 acting as a clock to perform the sampling.

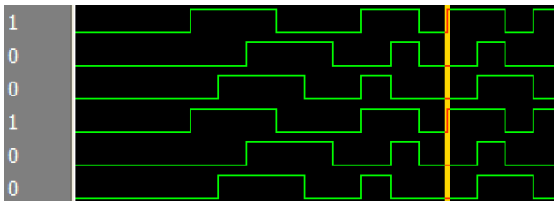
3.4.2 Transitional Sampling. Transitional sampling involves capturing the current value on multiple signal lines through the use of a shift register, as shown in Figure 3.8. The trigger to sample the signals is a transition, either $0 \rightarrow 1$ or $1 \rightarrow 0$. Figure 3.9 shows three different fingerprints created using the sample six signals with the sampling happening at different transitions on different signals. As with the NCS method, variations in the signals will result in different digital fingerprints.



(a) Sampled on second transition, signal six



(b) Sampled on third transition, signal six



(c) Sampled on third transition, signal 4

Figure 3.9: Transitional sampling performed on six signals.

IV. Test Circuits and Results

This chapter describes the test setups and results of the three proposed digital fingerprint methods when implemented on a Xilinx® Virtex-II Pro FPGA.

4.1 Digital Fingerprint Implementation on an FPGA

Due to time constraints, the digital fingerprint will be tested on using an FPGA rather and fabricating an ASIC. Testing will be done on a Xilinx® Virtex-II Pro XC2VP30 FPGA which is mounted to a Xilinx® Virtex-II Pro Development System board that is part of their Xilinx® University Program (XUP). A picture of the board can be seen in Figure 4.1.

The use of this board results in a more realistic test as not only is the FPGA being powered, but so is all the other components on the board. All of the circuits were designed structurally in VHDL and synthesized using the Xilinx® ISE.

4.2 Asynchronous LFSR Fingerprint Results

Initial tests with a VHDL coded 64-bit flip-flop LFSR running on the Virtex-II Pro FPGA showed correct functionality. When latches were used and the LFSR clocked, the output viewed by an attached logic analyzer showed all zeros. This presented a conundrum, as the LFSR was configured to maximal length with a non-zero seed, and therefore unable to enter into the all 0 state normally. Analysis performed on a 1-bit latch LFSR, with an XOR gate on the input and the feedback loop connected to one of the XOR inputs showed proper functionality. The addition of a second latch caused the all 0 state to reappear. The VHDL code was then implemented on an Altera Cyclone 2 FPGA to check if the problem was vendor specific. This resulted in a reoccurrence of the all 0 state. From these results, it was reasonable to conclude that a multiple latch feedback loop does not work across multiple CLBs due to the fundamental design of commercial FPGAs, likely a safety net of some sort that prevents non-stable signal transitions. This digital fingerprint generation method theoretically will work, but will have to be implemented as an ASIC.

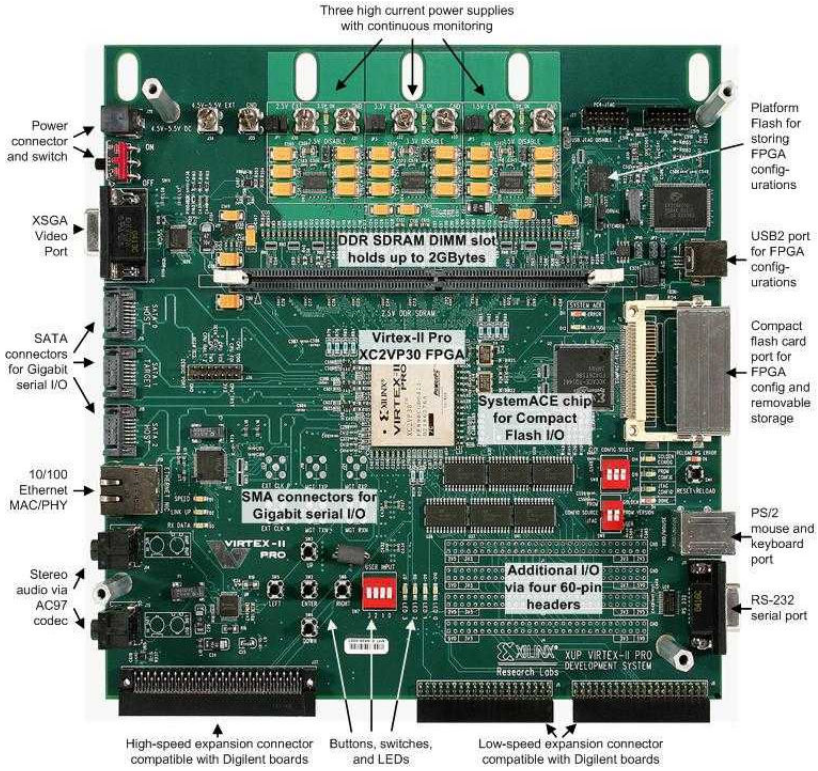
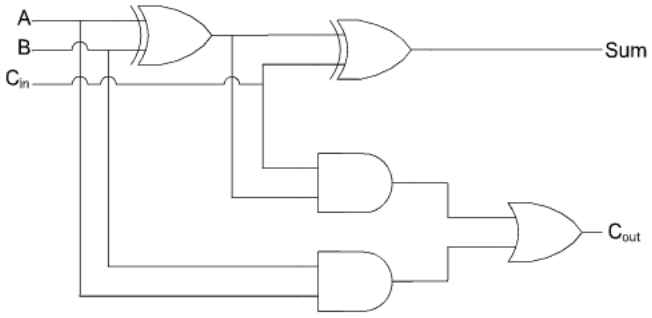


Figure 4.1: The XUP development board.

4.3 Transition Based Fingerprint Test Circuits

Any number of functionally complex circuits could be used to validate the transition based digital fingerprint methods. As design synthesis on to an FPGA results in combinational logic being implemented as LUTs, a functionally simple circuit can allow access to the LUTs just as well as a complex circuit. Therefore, a combinational multiplier will be utilized. This does not mean that the digital fingerprint is limited to the output of the circuit being examined. Any internal node that has signal transitions can be utilized in the creation of the digital fingerprint.

4.3.1 Combinational Multiplier. The 1-bit full adder, as shown in Figure 4.2, is used in beginning undergraduate digital logic classes to show the effect of gate delay on a circuit. Full adders are chained together to create ripple adders, where the carry-out of the n^{th} adder is connected to the carry-in of the $(n+1)^{th}$ adder. More significant bits are required to wait for carry values to propagate through before



(a)

A	B	C_{in}	Sum	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(b)

Figure 4.2: (a) The 1-bit full adder at the gate level. (b) Truth table of the full adder.

reporting a final result. In the mean time, intermediate values are placed on the outputs which then change when updated with the carry results. This causes the outputs to fluctuate, i.e. glitch, and perform multiple transitions between 0 and 1 until finally settling to a constant, final value. After modifying the full adder to perform partial product multiplication, as shown in Figure 4.3, multiple full adder chains can be combined to create a combinational multiplier, as shown in Figure 4.4. When reduced to LUTs, each full adder will become a pair of LUTs, one each for the sum and carry-out. Upon applying inputs, the sums and carrys are propagated throughout the array causing potentially a large number of transitions on the outputs. While the number of transitions that appear on the multiplier output will remain consistent across all FPGAs, the length of the transitions will change due to variations in the FPGA.

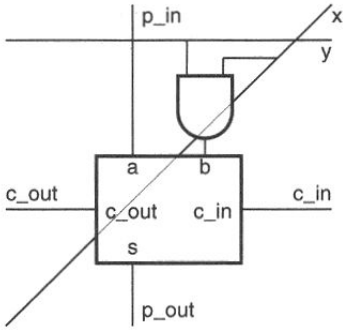


Figure 4.3: Partial product 1-bit full adder. The AND gate creates the partial product. [2]

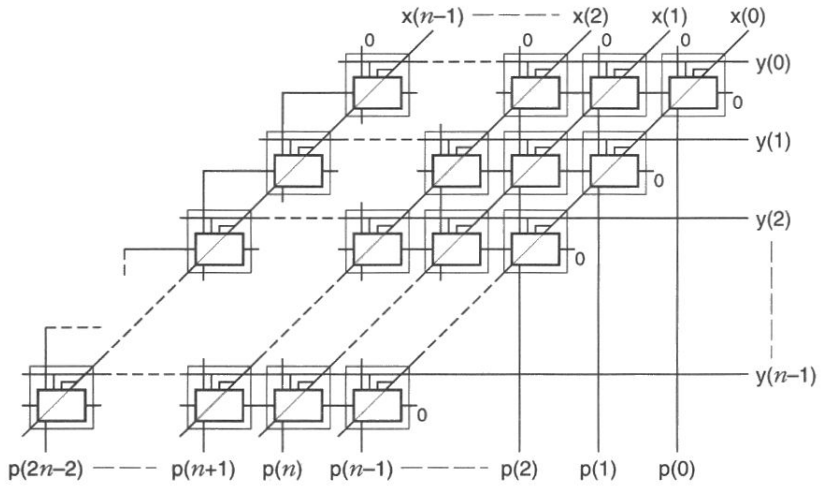


Figure 4.4: Full adder arrangement for creating a multiplier [2].

4.3.2 Test Circuits Input Via LFSR. Inputs for the test circuits are provided by an LFSR to minimize the input signal skew. Using external function or waveform generators would require a large amount of difficult synchronization of the inputs.

4.3.3 Test Circuits Limitations. Due to the implementation of positive edge triggered DFFs in the FPGA, only $0 \rightarrow 1$ transitions can be measured. While placing an inverter on the DFF clock input would allow examination of $1 \rightarrow 0$ transitions, it would also remove many of the short transitions that would allow the FPGAs to be differentiated. Also, inputs are provided to the multiplier test circuit via an LFSR internally, rather than an external input due to the lack of expansion header pins on the XUP board.

4.4 Nodal Cumulative Sampling

This circuit looks at the feasibility of counting transitions generated by the multiplier and the differences in their quantity across multiple FPGAs.

4.4.1 Structure. For this circuit, we will use a 32-bit combinational multiplier, also known as a 32x32 multiplier. Input values will be loaded via LFSR and the output of each bit connected to the clock input of a 20-bit wide one-hot-encoded shift register, as shown in Figure 4.5. A shift register is used instead of a binary counter due to the simpler implementation as the counter logic would require additional LUT overhead. As a transition occurs, the shift register will move a pre-loaded '1' from the LSB through the register to the MSB. The position that it stops on indicates the number of transitions seen. Only transitions with a long enough setup and hold time will activate a shift as shown in Figure 3.3. Naturally, this implies that due to limitations caused by the shift registers we may not be able to record all transitions, however more will be seen than if just using a logic analyzer directly. Additionally, the transitions that are counted are not only dependent on the delay variations of the LUTs but also on the variations of the DFFs used to create the shift registers.

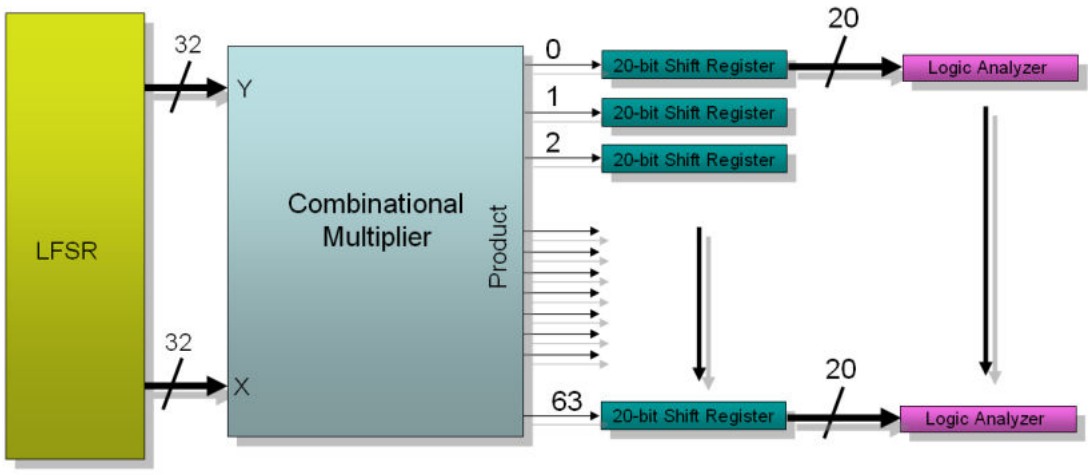


Figure 4.5: Structure of the nodal cumulative sampling circuit.

This will help create more variation between circuits that can be exploited for FPGA differentiation.

4.4.2 ModelSim Results. The circuit is structurally simulated in ModelSim using full adders to gain an understanding of the number of transitions that may be seen on the outputs. The simulation is limited by a number of factors: the minimum timestep allowed is 1 ns, there is no variation due to manufacturing in the delays of the full adders, the shift register latches in all data immediately upon a clock signal edge, and all interconnection wires are considered perfect. Due to the speed of the LUTs and the setup/hold times of the shift registers, it is likely that not all the transitions shown by the ModelSim output will actually be seen experimentally. Figure 4.6 shows a single bit of the multiplier output along with the resulting expected waveform of the shift register.

4.4.3 Multiplier Input Combinations. A multiplier input combination needs to provide fairly constant results over a large number of runs. Examination of ModelSim results for the combination multiplier show that the number of transitions increase near the center of the multiplier output.

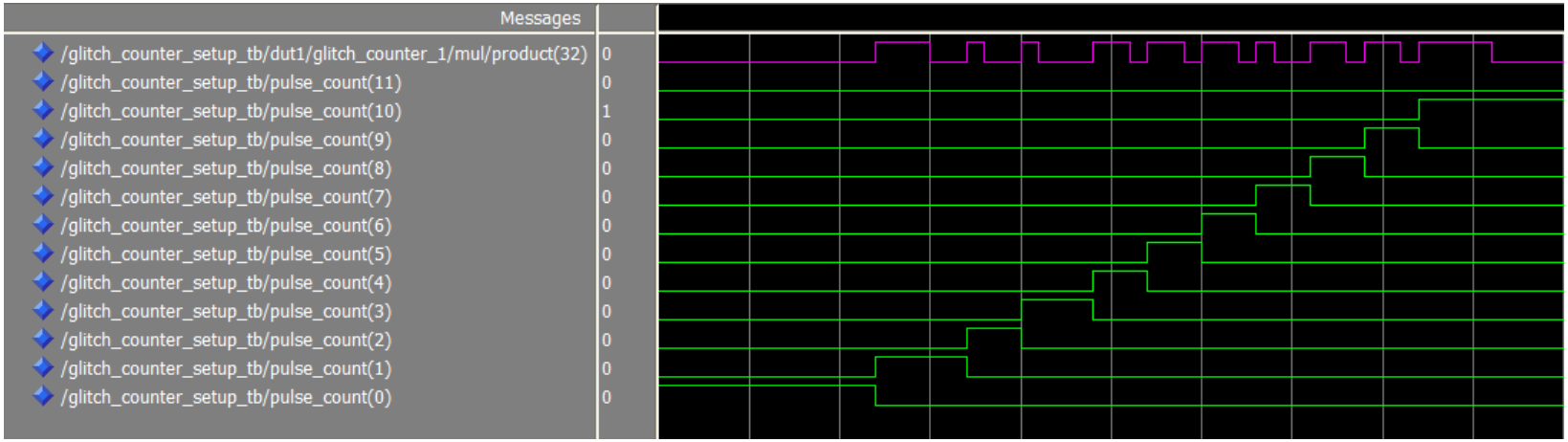


Figure 4.6: Multiplier output for bit 32 and predicted shift register output.

Implementing the transitional sampling circuit and using bit 32 as a gauge of the number of transitions, multiple input combinations are applied and output stability is examined. Results show that, when holding one multiplier input as 0xFFFFFFFF, output stability occurs when the second input, having n consecutive 1s, has $n + 1$ consecutive 0s following, i.e. 1100011000. Analysis of the full adder/LUT array shows that this is due to the structure of the array and the speed at which the array operates. Every row of the multiplier that reports a sum value of '1's potentially creates a set of transitions. Multiple rows of '1's cause both the sum and carry-out outputs to change, Figure 4.7. Too many simultaneous ripple adder rows of '1's causes a large number of fast changing transitions to appear at the output. These transitions violate the shift register clock input setup and hold times causing them not to be read. Rows of '0's act as a buffer by reducing the total number of transitions on the output and increasing the length of the remaining transitions. To make sure that output transitions can be consistently read, inputs of 0xFFFFFFFF and 0x49249149 are being used.

4.4.4 Nodal Cumulative Sampling Results.

4.4.4.1 General Observations. Data was collected on 10 FPGAs, for 64 bits per FPGA. Analysis of the data shows that the majority of multiplier outputs produce consistent transition counts over 100+ runs. The number of transitions on a particular output varies between different FPGAs, typically by a value of 1. In comparison with ModelSim predictions, the number of transitions are about half the predicted number. This is most likely due to the speed and width of the individual transition and the shift register setup and hold times. Bits 0-5 and 58-63 match the predicted values, most likely due to the fewer number of LUTs on these outputs, causing fewer transitions and allowing them to hold a 1 value longer so that they can be read by the shift registers. Bits 16-47 show the most activity with transition counts ranging between 4 to 6. Some multiplier outputs do not produce a constant number of transitions. These oscillating output values show constant transition counts on

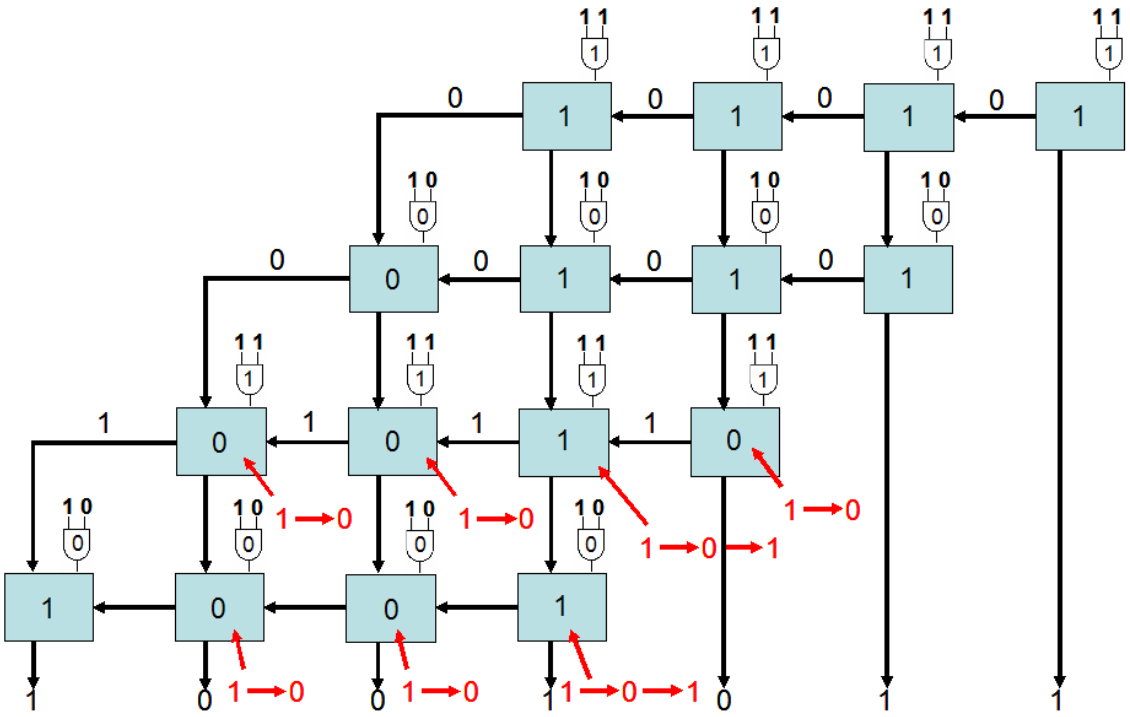


Figure 4.7: 4-bit combinational multiplier array showing propagation of sum and c_{out} for $15 * 5$ which are applied to the partial product AND gates on each adder. Red values show adder transitions due to delays.

other FPGAs. There appears to be no discernible pattern for which outputs show the oscillation.

4.4.4.2 Waveform Analysis. Examination of the shift register waveforms show several different phenomena that effect the final transition count. Some waveforms show a missed transition count as a shift register output does not show any activity, yet the next register reads '1' when clocked. This is due to sampling error by the logic analyzer due to the missing transition happening faster than the analyzer can sample. At other times the transition disappears. A shift register makes a transition from $1 \rightarrow 0$ showing the '1' has shifted, but the next register does not show the value. This can be attributed to a setup/hold time violation on the D input of the register. Figures 4.8(a) and 4.8(b) show these missed transitions. Similarly, the time multiple shift registers output constant '1's, Figure 4.8(c). This is also due

to setup/hold time violations, in this case for the clock line, therefore the lower '1' loaded register does not know to read its input value.

4.4.4.3 Nodal Cumulative Sampling Digital Fingerprint. As the number of transitions on an output change between FPGAs, these values can be used to create a digital fingerprint. Transition counts range from 0-6 suggesting a base-8 fingerprint. For a 32-bit multiplier output (64 bits), this then gives the theoretical maximum of unique fingerprints as:

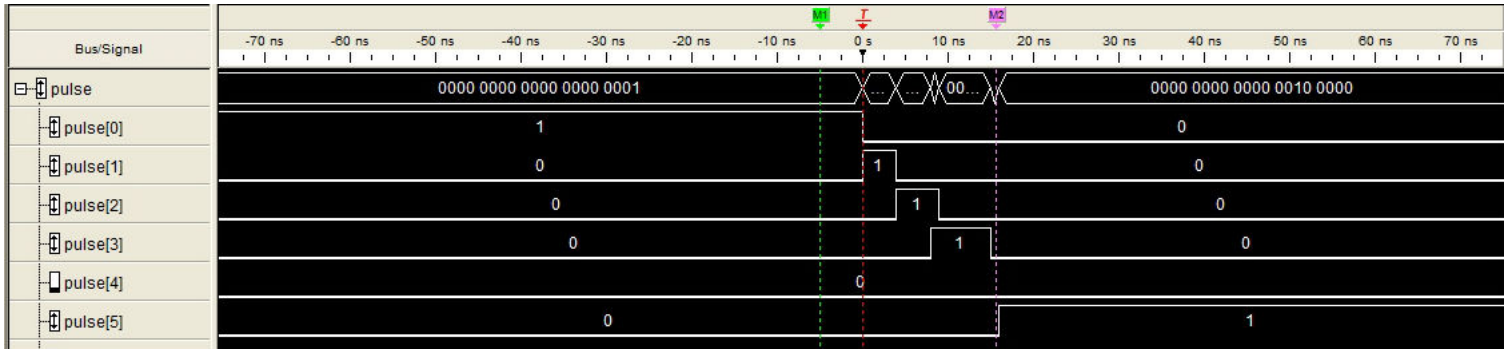
$$8^{64} \approx 6.28 * 10^{54}$$

Examination of the data shows that the actual number is less due to two constraints. First, a number of outputs, mainly focused in outputs 0-5 and 59-63, report the same value across all FPGAs. This is due to a lack of transitions on these signal lines, as the number of LUTs for feeding these outputs is lower than for other outputs due to the staggered nature of the ripple adders.

The second constraint is the number of outputs that show some oscillation of their transition count values. Depending on the severity of the oscillation, these outputs could still be used but a error correction factor for the fingerprints would have to be created to account for this activity. As an example of the effect of these constraints, Table 4.1 shows outputs 0-15 for all ten FPGAs. Only five of the outputs meet the constraints. Output 12, and others like it that show no change, may also meet the constraints with testing of additional FPGAs.

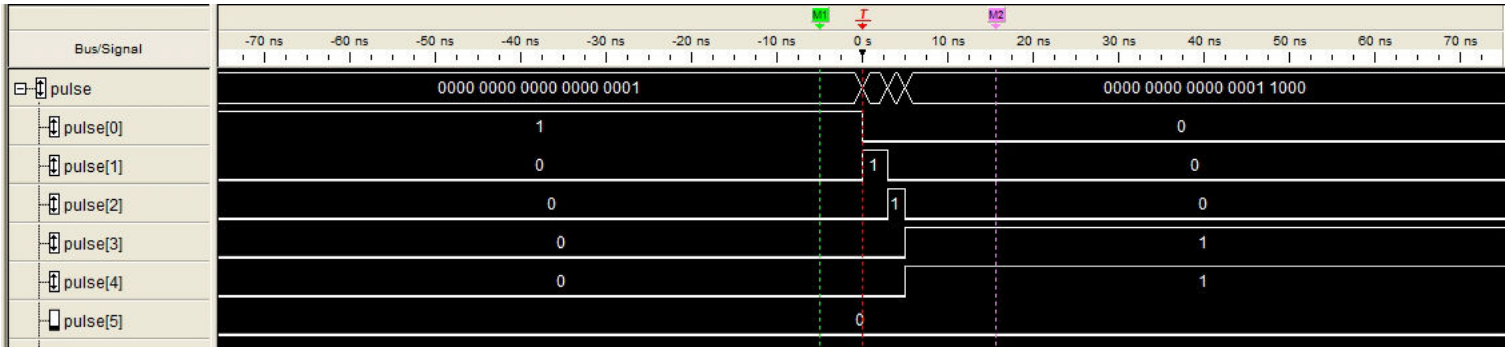
Applying these two constraints to the entire multiplier output results in 18 outputs that can be used to create a digital fingerprint. Table 4.2 shows the resulting fingerprints for the ten FPGAs. Each is unique and with using only 18 values, $8^{18} = 18,014,398,509,481,984$ FPGAs can be identified.

Because implementation of this digital fingerprint will be done on a binary system, conversion from base-8 to binary results in a three-fold increase in the length



(a) Missing transition due to sampling error.

(b) Missing transition due to setup/hold time violation.



(c) Multiple '1's due to setup/hold time violation.

Figure 4.8: Transition count waveforms with sampling errors.

Table 4.1: Transition counts for outputs 0-15.

FPGA	Pin_{15}	Pin_{14}	Pin_{13}	Pin_{12}	Pin_{11}	Pin_{10}	Pin_{9}	Pin_{8}	Pin_{7}	Pin_{6}	Pin_{5}	Pin_{4}	Pin_{3}	Pin_{2}	Pin_{1}	Pin_{0}
1	2	2	2	2	3	2	1	2	1,2,3	1	1	2	1	1	1	1
2	2	2	2	2	3	2	1	2	1,2,3	1	1	2	1	1	1	1
3	2	2	2	2	2	2	1	2	2	1	1	2	1	1	1	1
4	2	2	2	2	2	2	1	2	3	2	1	2	1	1	1	1
5	2	2	2	2	2	3	1	2	3	1	1	2	1	1	1	1
6	2	2	2	2	2	2	1	2	2	1	1	2	1	1	1	1
7	2	2	2	2	2	2	1	2	2	2	1	2	1	1	1	1
8	2	3	2	2	2	2	1	2	2	2	1	2	1	1	1	1
9	2,3	3	3	2	2	2	2	2	3	1,2	1	2	1	1	1	1
10	2,3	2	2	2	3	3	1	2	3	1,2	1	2	1	1	1	1

Table 4.2: Digital Fingerprints for the nodal cumulative sampling circuit FPGAs.

	Multiplier Outputs																	
FPGA	Pin_{51}	Pin_{48}	Pin_{47}	Pin_{46}	Pin_{42}	Pin_{35}	Pin_{34}	Pin_{30}	Pin_{27}	Pin_{25}	Pin_{21}	Pin_{19}	Pin_{17}	Pin_{14}	Pin_{13}	Pin_{11}	Pin_{10}	Pin_{9}
1	3	5	4	2	2	5	5	4	2	4	4	4	2	2	2	3	2	1
2	3	5	4	3	1	5	4	5	2	4	4	3	3	2	2	3	2	1
3	3	5	4	2	1	5	4	5	2	4	4	4	3	2	2	2	2	1
4	3	5	4	2	1	5	5	4	2	3	1	4	2	2	2	2	2	1
5	3	4	4	3	1	5	5	4	3	4	4	3	3	2	2	2	3	1
6	3	5	4	2	1	6	4	6	2	4	4	4	3	2	2	2	2	1
7	2	5	4	2	2	5	3	5	2	3	4	5	3	2	2	2	2	1
8	3	5	3	2	2	3	5	4	2	3	1	4	3	3	2	2	2	1
9	2	4	4	2	2	3	5	4	3	3	1	3	3	3	3	2	2	2
10	2	5	4	2	1	3	4	4	3	3	4	4	3	2	2	3	3	1

of the ID without altering the number of possible IDs. This allows 18 bits of the multiplier output to become a 54-bit ID or through restructuring of the ripple adders, the entire 64-bit output becomes a 192-bit value.

Regardless of which implementation is used, all ten FPGAs can be distinctly identified. The full 64-octal ID for each FPGA can be found in Appendix A.

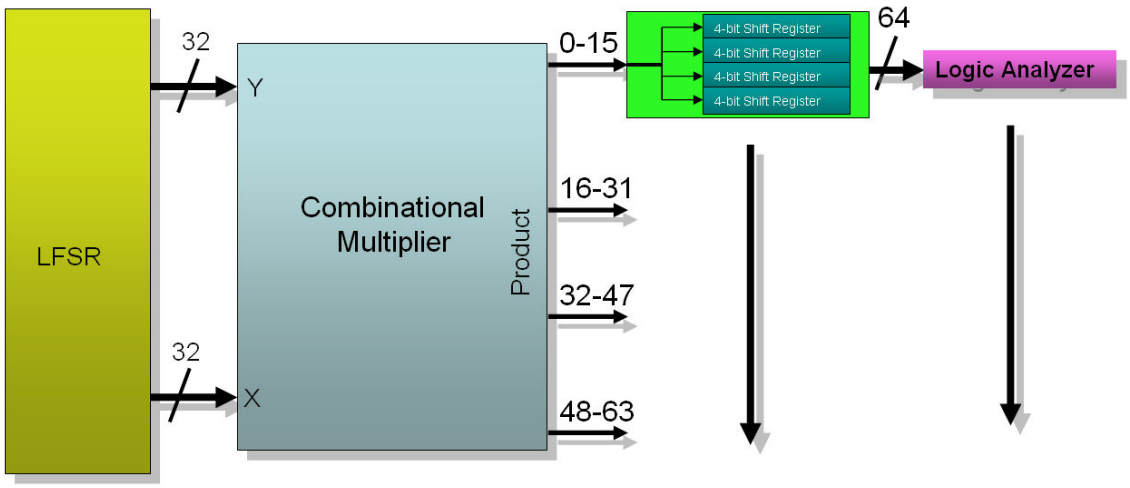


Figure 4.9: Structure of the transitional sampling circuit.

4.5 Tranistional Sampling

This circuit involves the capture of the multiplier output at pseudo-random times using a multiplier output as a pseudo-clock signal for the creation of a digital fingerprint.

4.5.1 Structure. The structure of this circuit is similar to the NCS circuit. An LFSR is attached to the input of the multiplier, while a shift register is placed on the output, as shown in Figure 4.9. In this case, the shift register is only 4 bits wide so as to capture only four samples of the output (n_0 to n_3). Of the multiplier outputs, nine were chosen to serve as the shift registers' clock input based on consistent transition counts. These were determined based on results from the first five FPGAs tested by the NCS circuit. Data collection on the second five FPGAs was performed simultaneously with this circuit. The chosen signals outputs are pins: 18, 19, 21, 24, 25, 28, 30, 34, and 40.

With four shift registers per output, 64 pins are required to read 16 output bits. Due to the limitations regarding the number of pins available on the FPGA board and the number of fly-leads available for the logic analyzer, collection of the data from the shift registers is broken into four 16-bit sections: bits 0-15, 16-31, 32-47, and 48-63.

Table 4.3: Output and clock errors for bits 47-32 of FPGA 5, Clk 25. commonly appearing values, clock error, and output error.

Output	Samples			
	n_0	n_1	n_2	n_3
1	0x9258	0x9055	0x1007	0x3a55
2	0x9258	0x921e	0x9055	0x1007
3	0x9208	0x9055	0x1007	0x3a55

4.5.2 ModelSim Results. ModelSim confirms the functionality of the transitional sampling circuit. Extrapolation from the results shows that changes in the transitions on the shift register clock input could result in radically different sample results.

4.5.3 Transitional Sampling Results.

4.5.3.1 General Observations. Testing of the transitional sampling circuit was done on twenty FGPAs, utilizing nine multiplier outputs serving as trigger lines to capture four samples per output. Analysis shows that the sample values are fairly consistent over multiple data capture runs for all FPGAs and output trigger signals. For a number of the output samples, variations in the data stored in the shift registers exist. These are caused by two different types of errors. The first is due either to outputs being on the edge of readability by the shift register or clock signal skew causing the capture to be off. This causes only a few bits to change over multiple runs. The second is due to the clock signal being read inconsistently by the shift registers resulting in a large number of bits changing over multiple runs. Both of these errors are due to the same underlying cause, namely the transition being on the edge of readability by the shift registers. Table 4.3 shows the samples of a single board with both output and clock based errors.

4.5.3.2 Transitional Sampling Digital Fingerprint. In order validate this digital fingerprint method, all twenty FPGAs must be differentiable across a single sample. This is done my performing a bitwise XOR between the sample values

Table 4.4: FPGA 16, Clk 25 w/ error factors

	Samples			
Output	n_0	n_1	n_2	n_3
1	0x9258	0x9055	0x1007	0x3a55
2	0x9258	0x921e	0x9055	0x1007
3	0x9208	0x9055	0x1007	0x3a55
Errors	2	5	13	15

across all the FPGAs for a particular clock signal. Any differences will result in a '1' value for that bit. Since errors are being seen in the samples, any differentiation between FPGAs must also take into account errors when the sample value is read. To do this, we create an error factor for each sample of each FPGA. This factor is the maximum number of bits that a sample is seen to change from its normal output, i.e. the one that shows up the most often. An output error or a clock error value is used as the error factor as they are mutually exclusive. In the case of no clear, consistent output, one was chosen at random from the multiple runs of data. Table 4.4 adds the error factors for the samples of the FPGA given in Table 4.3.

Table 4.5 summarizes the results showing FPGA differentiation for each clock signal and sample with full breakdown by bit in Appendix B. Also shown is the breakdown for the number of bits that varied and stayed the same between FPGAs for that clock along with the number of bits that had an error for at least one FPGA for that sample and clock.

Examination of these results shows that on average the best differentiation is achieved with sample n_2 despite the number of errors, with clock 24 correctly identifying all 20 FPGAs. The erratic differentiation values across all samples and clocks come from which bits have error, see Appendix B for the actual bit vectors and error locations. If the majority of error is on bits that should be the same between FPGAs, there is little effect on the differentiation. Based on the data, these cases happen when there are to few bits to differentiate FPGAs, thus the result is the same. When the error effects bits that are different between FPGAs, naturally the differentiation is greatly reduced.

Table 4.5: FPGA differentiation results by sample.

(a) n_0

Clk	#Differentiated	#of bits same	#of bits different	#errors
18	8	28	36	26
19	11	53	11	4
21	9	53	11	7
24	17	36	28	8
25	4	56	8	9
28	4	56	8	10
30	9	35	29	17
34	2	47	17	18
40	10	43	21	6

(b) n_1

Clk	#Differentiated	#of bits same	#of bits different	#errors
18	14	29	35	33
19	5	33	31	27
21	6	38	26	16
24	8	43	21	5
25	3	45	19	20
28	1	45	19	37
30	5	28	36	35
34	5	25	39	28
40	11	31	33	19

(c) n_2

Clk	#Differentiated	#of bits same	#of bits different	#errors
18	13	30	34	28
19	4	28	36	27
21	8	34	30	12
24	20	30	34	3
25	7	22	42	30
28	1	21	43	43
30	11	19	45	43
34	13	14	50	24
40	11	19	45	29

(d) n_3

Clk	#Differentiated	#of bits same	#of bits different	#errors
18	13	44	20	20
19	4	38	26	32
21	8	35	29	18
24	13	32	32	19
25	5	14	50	38
28	2	17	47	51
30	15	13	51	47
34	3	14	50	52
40	2	14	50	30

The creation of a digital fingerprint is dependent upon being able to have a repeatable value. The transitional sampling circuit is hard to control and requires a good clock signal that provides transitions that meet the shift register's flip-flops' setup and hold times. This circuit may be more applicable in conjunction with nodal cumulative sampling, by providing an asynchronous read of the transition count value.

4.5.3.3 Digital Fingerprinting Circuits Results vs Total Population. With both digital fingerprinting circuits, all FPGAs can be identified, 10/10 for the glitch count and 20/20 for the asynchronous capture. These numbers represent a fairly small number compared to the total population of Virtex-II Pro FPGAs currently available. In order to relate this success to the larger population, use of the hypergeometric distribution is required.

The hypergeometric distribution gives the probability that the total number of successes in a sample are representative of a larger population. To use this analogy, take a box filled with two colors of balls. A sample of the ball population shows that $x\%$ of the balls are of one color. Is that percentage representative of the whole population? Equation (4.1) gives the hypergeometric distribution, where N_p is the population size, k is the number of items with the desired characteristic, N_s is the total number of samples drawn, and x is the number of samples that have the desired characteristic.

$$p(x) = \frac{\left(\frac{k}{x}\right)\left(\frac{N_p-k}{N_s-x}\right)}{\left(\frac{N_p}{N_s}\right)} \tag{4.1}$$

Performing some mathematical manipulation, we can get the distribution in terms of percentages by extracting the population and sample sizes, given by Equation (4.2), with the following variables:

$$p(x) = \frac{\left(\frac{CN_p}{xN_s}\right)\left(\frac{(1-C)N_p}{(1-x)N_s}\right)}{\left(\frac{N_p}{N_s}\right)} \tag{4.2}$$

N_p	Total population
C	Unknown % of population with desired characteristic with range, $0 \leq x \leq 1$
CN_p	Total # in population with desired characteristic
N_s	Sample size
x	% of samples with desired characteristic, with range, $0 \leq x \leq 1$
xN_s	Total # in sample with desired characteristic

Using the assumption from Chapter 2 that there is a statistically significant variation in all FPGAs and therefore the entire population is different, then $C = 1$. As x is determined from the sample, we want to know its validity for the entire population, given the sample size. This is done by using the standard deviation of the sample, σ. To have high confidence in the result, i.e. 0.997 probability that the resulting values are correct, three times the standard deviation (3σ) is used. The result, shown in Equation (4.3), is a quadratic equation whose roots, when solved for C, give the upper and lower bounds for the probability that the measured x value from the sample is correct. The closer the roots are to 1, the more accurately our differentiation method works for all cases. Note that λ is the confidence level of the result and $\lambda = 3$ for 3σ.

$$(x - C)^2 = \lambda^2 \frac{C(1 - C)}{N_s} \tag{4.3}$$

For the NCS circuit, C is solved when $x = 10$ and $N_s = 10$ giving upper and lower bounds of 1.0000 and 0.5263, respectively. As the worse case needs to be always considered, therefore, there is a 52.63% probability that the glitch counting method in its current configuration would be able to correctly differentiate between all Virtex-II Pro FPGAs. Solving for the transitional sampling circuit with $x = 20$ and $N_s = 20$ gives a 68.97% correct differentiation. As the hypergeometric distribution is significantly dependent on the sample size, the small sample sizes used in this

testing can not provide high confidence results. To achieve 90%, 95%, and 99% confidence, sample sizes of 81, 171, and 887 FPGAs respectively, assuming complete differentiation, would have to be used. The over 50% confidence for the two digital fingerprint circuits shows they provide better results than just guessing.

V. Conclusions and Future Work

This chapter contains conclusions to this research effort and their significance. Future research recommendations are also included for continued efforts.

5.1 Conclusions of Research

With the increase of COTS usage in military applications, the probability that critical functionality will be reverse engineered or tampered with becomes high. The creation of a digital fingerprint that can provide a unique identifier that can both identify a circuit and potentially grant functionality lock-in to a specific piece of silicon becomes a critical necessity in securing the United States' technological edge.

This research proposed that variations in semiconductor fabrication have a measurable effect and can be used in conjunction with circuit functionality to create a digital fingerprint. Three methods of creating this digital fingerprint were proposed, asynchronous LFSR, nodal cumulative sampling, and transitional sampling, of which the latter two were able to be implemented on an FPGA.

Test circuit functionality was implemented on the FPGA to gain access to the LUTs directly in order to properly test the digital fingerprint methods. The resulting data shows variations in the number of transitions of the output bits across multiple FPGAs. The variations are for the most part consistent and provide a good foundation to allow the creation of a digital fingerprint. The two fingerprint methods implemented, using the number of transitions directly as a fingerprint and using the transitions to sample signals asynchronously, both resulted in successful identification of all FPGAs.

5.2 Future Research

There are a number of different follow-on research topics that should be investigated detailed below, several of which may be thesis topics in themselves.

5.2.1 Other Register Types. Due to the limitations of the FPGA used, only $0 \rightarrow 1$ transitions were examined. Through different implementations of flip-flops, most likely ASIC, $1 \rightarrow 0$ transitions, 1-level and 0-level pulses, or some combination of all four can be used to create a digital fingerprint

5.2.2 ASIC Asynchronous LFSR. The implementation of the asynchronous LFSR as an ASIC design need to be done to verify it's feasibility. While it should work in theory, the restrictions of the FPGA architecture do not allow its creation.

5.2.3 LFSR Based Nodal Cumulative Sampling and Transitional Sampling. Instead of using the internal nodes of circuit to create the digital fingerprint, a LFSR could be used as part of a stand alone digital fingerprint circuit. The LFSR is capable of generating pseudo-random outputs, based on tap position, seed value, and number of clock cycles operated. Attaching the LFSR to another fingerprint generating circuit, like the NCS or transitional sampling, would result in a much more scrambled fingerprint. One step further would be to use the proposed asynchronous LFSR.

5.2.4 Digital Fingerprint Encryption/Decryption Key. Because of the uniqueness of the digital fingerprint, it could be used in the creation of an encryption/decryption key set. Data would have to undergo these processes on the FPGA as to keep the keys secret. This could also be applied to the FPGA bitstream itself, allowing it to be encrypted in such a way that only one FPGA could use it.

5.2.5 Stress Testing. An in depth look at the long term stability of digital fingerprints proposed in this thesis must be done. As an FPGA or ASIC's characteristics can change over time, the effect this has on the fingerprint needs to be examined.

Table A.1: Cumulative signal transitions outputs by pin

FPGA	$P_{in}63$	$P_{in}62$	$P_{in}61$	$P_{in}60$	$P_{in}59$	$P_{in}58$	$P_{in}57$	$P_{in}56$	$P_{in}55$	$P_{in}54$	$P_{in}53$	$P_{in}52$	$P_{in}51$	$P_{in}50$	$P_{in}49$	$P_{in}48$	$P_{in}47$	$P_{in}46$	$P_{in}45$	$P_{in}44$	$P_{in}43$	$P_{in}42$
1	0	1	1	1	2	2	1	2	2	2	2	4	3	3	5	5	4	2	2	4	3	2
2	0	1	1	1	2	1	1	2	1	2	2	4	3	3	5	5	4	3	3	4	3	1
3	0	1	1	1	2	1,2	1	2	1	2	2	4	3	3	5	5	4	2	3	4	2	1
4	0	1	1	1	2	2	1	2	1	2	2	4	3	3	5	5	4	4	2,3	4	2,3	1
5	0	1	1	1	2	2	1	2	1	2	2	4	3	3	5	5	4	4	3	4	3	1
6	0	1	1	1	2	2	1	2	1	2	2	4	3	3	5	5	4	4	3	4	2	1
7	0	1	1	1	2	2	1	2	1	2	2	3	2	3	4	5	4	2	4	4	3	2
8	0	1	1	1	2	1	1	2	1	2	2	3,4	3	3	5	5	3	2	3	4	2	2
9	0	1	1	1	2	2	1	2	1,2	2	2	3,4	2	3	4,5	4	4	2	1	4	3	2
10	0	1	1	1	2	2	1	2	1	2	2	3	2	3	4,5	5	4	2	2,3	4	2	1

FPGA	$P_{in}41$	$P_{in}40$	$P_{in}39$	$P_{in}38$	$P_{in}37$	$P_{in}36$	$P_{in}35$	$P_{in}34$	$P_{in}33$	$P_{in}32$	$P_{in}31$	$P_{in}30$	$P_{in}29$	$P_{in}28$	$P_{in}27$	$P_{in}26$	$P_{in}25$	$P_{in}24$	$P_{in}23$	$P_{in}22$	$P_{in}21$
1	4	3	3	4	3	4	5	5	5	4	4,5	4	5	5	2	4	4	4	5	3	4
2	4	4	3	4	4	5	5	4	5	4	5	5	5	5	2	4	4	4	5	3	4
3	4	4	3	4	3,4	5	5	4	4,5	4	6	5	5,6	4	2	2,3	4	3	4,5	2	4
4	4	4	3	4	3,4	3,4	5	5	5,6	4,5	6	4	5	5	2	4	3	3	4,5	2,3	1
5	4	4	3	4	4	4	5	5	2	5	6	4	6	5	3	4	4	4	5	3	4
6	4	4	3	4	5	5	6	4	5	4	6	6	5	5	2	4	4	4	5	2	4
7	4	3	3	4	4	3	5	3	5	4	6	5	5	5	2	4	3	3,4	4	3,4	4
8	4	3	3	4	5	3	3	5	3	4	6	4	4	4,5	2	4	3	3	5	3	1
9	4	3,4	3	3,4	3	3	3	5	5	2,3	6	4	4	3,4	3	4	3	3,4	5	3,4,5	1
10	4	3	3	4	3	4	3	4	4	4	6	4	4,5,6	3,4,5	3	4	3	3	5	4	4

FPGA	$P_{in}20$	$P_{in}19$	$P_{in}18$	$P_{in}17$	$P_{in}16$	$P_{in}15$	$P_{in}14$	$P_{in}13$	$P_{in}12$	$P_{in}11$	$P_{in}10$	$P_{in}9$	$P_{in}8$	$P_{in}7$	$P_{in}6$	$P_{in}5$	$P_{in}4$	$P_{in}3$	$P_{in}2$	$P_{in}1$	$P_{in}0$
1	3	4	3	2	3	2	2	2	2	3	2	1	2	1,2,3	1	1	2	1	1	1	1
2	3	3	4	3	3	2	2	2	2	3	2	1	2	1,2,3	1	1	2	1	1	1	1
3	3	4	4	3	4	2	2	2	2	2	2	1	2	2	1	1	2	1	1	1	1
4	3	4	3	2	3	2	2	2	2	2	2	1	2	3	2	1	2	1	1	1	1
5	3	3	4	3	3	2	2	2	2	2	3	1	2	3	1	1	2	1	1	1	1
6	3	4	4	3	4	2	2	2	2	2	2	1	2	2	1	1	2	1	1	1	1
7	3,4	5	4	3	2,3,4	2	2	2	2	2	2	1	2	2	2	1	2	1	1	1	1
8	3	4	3,4	3	2,3	2	3	2	2	2	2	1	2	2	2	1	2	1	1	1	1
9	3	3	3	3	2,3,4	2,3	3	3	2	2	2	2	2	3	1,2	1	2	1	1	1	1
10	3	4	3	3	2,3	2,3	2	2	2	3	3	1	2	3	1,2	1	2	1	1	1	1

Table B.1: FPGA outputs for Sample 0, Clk 18

FPGA	Pin$_{0}$	Pin$_{1}$	Pin$_{2}$	Pin$_{3}$	Pin$_{4}$	Pin$_{5}$	Pin$_{6}$	Pin$_{7}$	Pin$_{8}$	Pin$_{9}$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0
2	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	0	0	0	1	0	0	0	0	0	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0
3	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0
4	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0
5	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1	1	0	1	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
6	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
7	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	0	0	1	1	0	0	0	0	0	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
8	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	1	0	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
9	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	1	0	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
10	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0
11	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
12	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	1	1	0	1	0	1	1	0	0	0	1	0	1	0	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
13	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	0	0	0	1	0	0	0	0	0	0	1	0	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
14	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
15	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	0	0	0	1	0	0	0	0	0	0	1	0	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
16	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	1	0	1	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
17	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
18	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	0	0	1	1	0	0	0	0	0	0	1	0	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
19	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
20	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	1	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0

Table B.2: FPGA outputs for Sample 0, Clk 19

FPGA	Pin$_{0}$	Pin$_{1}$	Pin$_{2}$	Pin$_{3}$	Pin$_{4}$	Pin$_{5}$	Pin$_{6}$	Pin$_{7}$	Pin$_{8}$	Pin$_{9}$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	1	1	1	0	1	0	0	1	1	1	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
3	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
4	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
5	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	1	1	1	0	1	0	0	1	1	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
7	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
8	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
9	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
10	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	1	1	0	0	1	0	0	1	1	1	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
12	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	1	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
13	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
14	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
15	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
16	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	1	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
17	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	1	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
18	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
19	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
20	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0

Table B.3: FPGA outputs for Sample 0, Clk 21

FPGA	Pin$_0$	Pin$_1$	Pin$_2$	Pin$_3$	Pin$_4$	Pin$_5$	Pin$_6$	Pin$_7$	Pin$_8$	Pin$_9$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	1	0	1	0	1
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	1	0	1	0	1
3	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	1	1	0	0	0	0	1
4	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	1	0	1	0	1
5	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	1	0	1	0	1
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	0	1
7	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	1	0	1	0	1
8	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	1	0	1	0	1
9	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	1	0	1	0	1
10	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	1	0	1	0	1
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	1
12	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	1	0	1	0	1
13	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	1	0	1	0	1
14	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	1	0	1	0	1
15	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	0	0	1	0	1
16	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	1	0	1	0	1
17	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	1	0	0	0	1
18	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	1
19	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	1	0	1	0	1
20	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0

FPGA	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	1	1	1	0	0	0	1	1	0	0	0	0	1	0	0	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0
2	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
3	1	1	1	0	0	0	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
4	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
5	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
6	1	1	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
7	1	1	1	0	0	0	1	1	0	0	0	0	1	0	0	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0
8	1	1	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
9	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0
10	1	1	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
11	1	1	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0
12	1	1	1	0	0	0	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
13	1	1	1	0	0	0	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
14	1	1	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
15	1	1	1	0	0	0	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
16	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0
17	1	1	1	0	0	0	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
18	1	1	1	0	0	0	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
19	1	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
20	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0

Table B.4: FPGA outputs for Sample 0, Clk 24

FPGA	Pin$_0$	Pin$_1$	Pin$_2$	Pin$_3$	Pin$_4$	Pin$_5$	Pin$_6$	Pin$_7$	Pin$_8$	Pin$_9$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	0	0	0	1	0	1	0	0	1	1	1
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	1	1	0	0	0	0	1	0	1	0	0	1	1	0
3	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	1	1	0	0	0	0	1	0	1	0	0	1	1	1
4	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	1	1	0	0	0	0	1	0	1	0	0	1	1	1
5	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	0	1	0	1	0	1	0	0	1	1	1
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	1	1	0	0	0	0	1	0	1	0	0	1	1	1
7	1	1	1	0	1	1	0	1	1	0	1	1	1	1	0	0	1	1	1	0	0	0	0	0	1	0	1	0	0	1	1	1
8	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	1	1	0	0	0	0	1	0	1	0	0	1	1	1
9	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	0	1	0	1	0	1	0	0	1	1	1
10	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	0	0	0	1	0
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	1	1	0	0	1	0	1	0	1	0	0	1	1	1
12	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	0	1	0	0	0	0	1	1	1	0	0	0	1	0
13	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	0	0	0	1	0	1	0	0	1	1	1
14	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	1	1	0	0	0	0	1	0	1	0	0	1	1	1
15	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	0	0	0	1	0	1	0	0	1	1	1
16	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	1	1	0	0	0	0	1	0	1	0	0	1	1	1
17	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	1	1	0	0	0	0	1	0	1	0	0	1	1	1
18	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	1	1	0	0	0	0	1	0	1	0	0	1	1	1
19	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	0	1	0	0	0	0	1	1	1	0	0	0	1	0
20	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	0	0	0	1	0

FPGA	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
2	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
3	1	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0
4	1	0	0	0	0	1	1	0	0	0	0	1	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
5	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
6	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0
7	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
8	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
9	1	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
10	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
11	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0
12	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
13	1	0	0	0	0	1	1	0	0	0	0	1	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
14	1	0	0	0	1	1	1	0	0	0	0	1	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
15	1	0	0	0	1	1	1	0	0	0	0	1	1	0	1	0	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0
16	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
17	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0
18	1	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0
19	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
20	1	1	0	0	1	0	1	0	0	1	0	1	0	1	0	0	0	1	1	0	1	0	0	0	1	0	0	1	0	0	1	0

Table B.5: FPGA outputs for Sample 0, Clk 25

FPGA	Pin_{0}	Pin_{1}	Pin_{2}	Pin_{3}	Pin_{4}	Pin_{5}	Pin_{6}	Pin_{7}	Pin_{8}	Pin_{9}	Pin_{10}	Pin_{11}	Pin_{12}	Pin_{13}	Pin_{14}	Pin_{15}	Pin_{16}	Pin_{17}	Pin_{18}	Pin_{19}	Pin_{20}	Pin_{21}	Pin_{22}	Pin_{23}	Pin_{24}	Pin_{25}	Pin_{26}	Pin_{27}	Pin_{28}	Pin_{29}	Pin_{30}	Pin_{31}	Pin_{32}	Pin_{33}	Pin_{34}	Pin_{35}	Pin_{36}	Pin_{37}	Pin_{38}	Pin_{39}	Pi_{n40}	Pin_{41}	Pin_{42}	Pin_{43}	Pin_{44}	Pin_{45}	Pin_{46}	Pin_{47}	Pin_{48}	Pin_{49}	Pin_{50}	Pin_{51}	Pin_{52}	Pin_{53}	Pin_{54}	Pin_{55}	Pin_{56}	Pin_{57}	Pin_{58}	Pin_{59}	Pin_{60}	Pin_{61}	Pin_{62}	Pin_{63}
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
3	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
4	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
5	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
7	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
8	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
9	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
10	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
12	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
13	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
14	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
15	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
16	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
17	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
18	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
19	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
20	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	1	1	1	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0

Table B.6: FPGA outputs for Sample 0, Clk 28

FPGA	Pin_{0}	Pin_{1}	Pin_{2}	Pin_{3}	Pin_{4}	Pin_{5}	Pin_{6}	Pin_{7}	Pin_{8}	Pin_{9}	Pin_{10}	Pin_{11}	Pin_{12}	Pin_{13}	Pin_{14}	Pin_{15}	Pin_{16}	Pin_{17}	Pin_{18}	Pin_{19}	Pin_{20}	Pin_{21}	Pin_{22}	Pin_{23}	Pin_{24}	Pin_{25}	Pin_{26}	Pin_{27}	Pin_{28}	Pin_{29}	Pin_{30}	Pin_{31}	Pin_{32}	Pin_{33}	Pin_{34}	Pin_{35}	Pin_{36}	Pin_{37}	Pin_{38}	Pin_{39}	Pi_{n40}	Pin_{41}	Pin_{42}	Pin_{43}	Pin_{44}	Pin_{45}	Pin_{46}	Pin_{47}	Pin_{48}	Pin_{49}	Pin_{50}	Pin_{51}	Pin_{52}	Pin_{53}	Pin_{54}	Pin_{55}	Pin_{56}	Pin_{57}	Pin_{58}	Pin_{59}	Pin_{60}	Pin_{61}	Pin_{62}	Pin_{63}
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	1	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
3	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
4	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
5	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	1	0	1	1	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
7	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	1	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
8	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
9	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
10	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	1	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
12	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	1	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
13	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
14	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
15	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
16	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
17	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	1	0	1	1	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
18	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	1	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
19	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
20	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0

Table B.7: FPGA outputs for Sample 0, Clk 30

FPGA	Pin$_{0}$	Pin$_{1}$	Pin$_{2}$	Pin$_{3}$	Pin$_{4}$	Pin$_{5}$	Pin$_{6}$	Pin$_{7}$	Pin$_{8}$	Pin$_{9}$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	1	0	1	0	1	0	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0	1	0	0	1	0	0	1	0
3	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
4	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	1	0	0	0	1	0	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
5	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	1	0	1	0	1	0	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	1	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	0	1	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
7	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	1	0	0	0	1	0	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
8	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	1	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
9	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	1	1	1	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0	1	0	0	1	0	0	1	0
10	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	0	0	0	1	0	0	0	0	1	0	1	1	1	1	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
12	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	1	1	0	0	1	0	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
13	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	1	1	0	0	0	0	0	1	1	1	0	0	1	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
14	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	0	0	0	1	0	1	0	1	0	0	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
15	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
16	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
17	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	0	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0	1	0	0	1	0	0	1	0
18	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	0	0	0	1	1	0	0	1	0	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0	1	0	0	1	0	0	1	0
19	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
20	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0

Table B.8: FPGA outputs for Sample 0, Clk 34

FPGA	Pin$_{0}$	Pin$_{1}$	Pin$_{2}$	Pin$_{3}$	Pin$_{4}$	Pin$_{5}$	Pin$_{6}$	Pin$_{7}$	Pin$_{8}$	Pin$_{9}$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
3	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
4	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
5	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
7	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
8	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
9	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
10	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
12	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
13	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
14	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
15	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
16	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
18	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
19	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
20	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0

Table B.9: FPGA outputs for Sample 0, Clk 40

FPGA	Pin$_{0}$	Pin$_{1}$	Pin$_{2}$	Pin$_{3}$	Pin$_{4}$	Pin$_{5}$	Pin$_{6}$	Pin$_{7}$	Pin$_{8}$	Pin$_{9}$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
3	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	1	1	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
4	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
5	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
7	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
8	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
9	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
10	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
12	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
13	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
14	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
15	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
16	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
17	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
18	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	1	0	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
19	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
20	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0

Table B.10: FPGA outputs for Sample 1, Clk 18

FPGA	Pin$_{0}$	Pin$_{1}$	Pin$_{2}$	Pin$_{3}$	Pin$_{4}$	Pin$_{5}$	Pin$_{6}$	Pin$_{7}$	Pin$_{8}$	Pin$_{9}$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	0	0	1	1	0	1	1	1	0	0	0	0	1	0	1	1	0	1	1	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
4	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	1	1	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0
5	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	0	0	0	0	1	1	0	1	0	1	0	1	1	0	0	1	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0
6	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	0	0	1	0	0	1	0	1	1	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0
7	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0
8	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	0	0	1	1	0	1	1	1	0	0	0	0	1	0	1	1	0	1	1	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
9	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	1	1	1	1	1	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
10	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	1	1	1	0	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0
12	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	1	1	1	0	0	0	1	1	0	1	0	0	1	0	1	0	0	1	1	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
13	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	1	1	0	0	0	0	0	1	1	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0
14	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	0	0	1	1	0	1	1	1	0	0	0	0	1	0	0	1	0	1	1	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
15	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0
16	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	1	1	1	1	1	0	0	0	0	1	0	1	1	1	1	0	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
17	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	0	0	1	1	0	1	1	1	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0
18	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	0	0	1	1	0	1	1	1	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
19	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
20	1	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	1	0	0	1	0	0	1	0	1	1	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0

Table B.11: FPGA outputs for Sample 1, Clk 19

FPGA	Pin$_0$	Pin$_1$	Pin$_2$	Pin$_3$	Pin$_4$	Pin$_5$	Pin$_6$	Pin$_7$	Pin$_8$	Pin$_9$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	0	1	1	0	0	1	1	0	0	0	0	1	0	0	1	1	1	1	0	1	0	0	0	1	0	1	0	1	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
2	1	1	1	0	1	1	0	1	1	0	1	0	1	0	1	1	1	1	0	0	1	0	0	1	1	0	0	1	1	1	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
3	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	1	1	0	0	1	1	0	0	1	0	0	1	0	1	0	0	1	1	1	1	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
4	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	0	1	1	1	0	1	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
5	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	1	0	0	0	1	1	0	1	1	0	0	1	0	0	1	1	1	1	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
6	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	0	1	1	0	0	1	1	0	0	0	0	1	0	0	1	1	1	1	1	1	0	0	0	1	0	1	0	1	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
7	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	0	1	1	0	0	1	1	0	0	0	0	1	0	0	1	1	1	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
8	1	1	1	0	1	1	0	1	1	0	1	0	1	0	1	1	1	1	0	0	1	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0
9	1	1	1	0	1	1	0	1	1	0	1	0	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	1	0	0	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
10	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	1	1	0	0	1	0	0	0	1	0	0	1	1	1	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
11	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	1	1	0	0	1	0	0	1	1	0	1	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
12	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	1	1	0	0	1	0	0	0	1	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
13	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	1	1	0	0	1	1	0	0	1	0	1	1	1	0	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
14	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	1	0	0	0	1	1	0	0	1	0	0	1	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
15	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	1	1	0	0	1	1	0	0	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
16	1	1	1	0	1	1	0	1	1	0	1	0	1	0	1	1	1	1	0	1	1	0	0	0	1	0	0	1	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
17	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	1	1	0	0	1	0	0	0	1	0	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
18	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	1	0	0	0	1	1	1	1	1	0	1	1	0	0	1	1	1	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
19	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	1	0	1	1	0	0	1	1	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
20	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	1	1	0	0	1	1	0	0	1	0	0	1	0	0	1	1	1	1	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0

Table B.12: FPGA outputs for Sample 1, Clk 21

FPGA	Pin$_0$	Pin$_1$	Pin$_2$	Pin$_3$	Pin$_4$	Pin$_5$	Pin$_6$	Pin$_7$	Pin$_8$	Pin$_9$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	0	0	1	0	1	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0
3	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0
4	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	1	0	0	0	0	1	1	0	1	0	1	1	0	1	0	1	0	0	0	0	0	1	1	0	0	0	0	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0
5	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0
7	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	1	0	0	0	0	1	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0
8	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	0	0	1	0	1	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0
9	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0
10	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0
12	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	0	0	1	0	1	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0
13	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	1	0	0	0	0	1	1	0	1	0	1	0	0	1	0	1	0	0	0	0	0	1	1	0	0	0	0	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0
14	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0
15	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	1	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	1	0	0	0	1	0	0
16	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	0	0	0	0	1	0	1	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	1	0	0	0	1	0	0
17	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	0	1	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0
18	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	1	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0
19	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	1	1	0	0	0	1	0	0
20	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	1	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0	1	0	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0

Table B.13: FPGA outputs for Sample 1, Clk 24

FPGA	Pin$_0$	Pin$_1$	Pin$_2$	Pin$_3$	Pin$_4$	Pin$_5$	Pin$_6$	Pin$_7$	Pin$_8$	Pin$_9$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0
3	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0
4	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0
5	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0
7	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0
8	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0
9	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0
10	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	0	1	1	1	0	1	0	0	1	0	0	1	1	0	1
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	1	0	1	0
12	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	0	1	1	1	0	1	0	1	1	0	0	1	1	0	1
13	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0
14	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0
15	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0
16	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0
17	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0
18	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0
19	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	1	1	1	0	1	0	1	1	0	0	0	1	0	1
20	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	0	0	1	1	1	0	1	1	1	1	0	0	1	1	0	0

FPGA	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	1	1	0	0	1	0	1	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
2	1	1	1	0	1	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
3	1	1	0	0	1	0	1	0	0	1	0	1	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
4	1	1	0	0	1	0	1	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
5	1	1	0	0	1	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
6	1	1	0	0	1	0	1	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
7	1	1	1	0	1	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
8	1	1	1	0	1	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
9	1	1	0	0	1	0	1	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
10	1	1	1	0	1	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
11	1	1	0	0	1	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
12	1	1	1	0	1	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
13	1	1	0	0	1	0	1	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
14	1	1	1	0	1	0	1	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
15	1	1	0	0	1	0	1	1	0	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
16	1	1	1	0	1	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
17	1	1	0	0	1	0	1	0	0	1	0	0	1	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
18	1	1	0	0	1	0	1	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
19	1	1	1	0	1	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0
20	1	1	0	0	1	0	1	1	0	0	0	1	1	1	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0

Table B.14: FPGA outputs for Sample 1, Clk 25

FPGA	Pin$_0$	Pin$_1$	Pin$_2$	Pin$_3$	Pin$_4$	Pin$_5$	Pin$_6$	Pin$_7$	Pin$_8$	Pin$_9$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	1	0	0
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	1	1	1
3	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	1	1	1	0	0	0	0	1	1	1	1	1	1
4	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	1	0	0
5	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	0	1	1	0	1	0	0
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	1	0	1
7	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	1	0	1
8	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	0	1	0	0	1	1
9	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	1	1	1	0	0	0	0	0	0	1	0	0	1
10	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	0	0	0	0	1	1	1	1	1
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	1	1	1	1	0	0
12	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	0	0	1
13	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	1	1	1
14	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	0	0	1
15	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	0	1	1
16	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	1	1
17	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	1	0	1
18	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	1	1	0	1	0	1
19	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	0	0	0	0	1	1	1	1	1
20	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	1	1	0	1	1	1

FPGA	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	0	0	1	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
2	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
3	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
4	1	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
5	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
6	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
7	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
8	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
9	0	0	1	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
10	0	0	1	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
11	0	0	1	1	1	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
12	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
13	0	0	0	0	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
14	1	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
15	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
16	1	0	1	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
17	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
18	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
19	1	0	1	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
20	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0

Table B.15: FPGA outputs for Sample 1, Clk 28

FPGA	Pin$_0$	Pin$_1$	Pin$_2$	Pin$_3$	Pin$_4$	Pin$_5$	Pin$_6$	Pin$_7$	Pin$_8$	Pin$_9$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	1	1	0	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
3	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	1	1	0	0	1	1	0	0	0	0	1	0	0	0	1	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
4	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	1	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
5	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	1	0	0	0	1	1	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
7	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	1	1	0	0	1	1	0	1	1	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
8	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	1	0	0	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
9	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	1	1	0	0	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
10	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	0	1	0	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	1	1	0	0	1	1	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0
12	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
13	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	1	0	0	1	0	0	0	1	1	1	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
14	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
15	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
16	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	1	0	0	1	0	1	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
17	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
18	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0
19	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	1	1	1	0	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
20	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	1	0	0	0	1	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0

Table B.16: FPGA outputs for Sample 1, Clk 30

FPGA	Pin$_0$	Pin$_1$	Pin$_2$	Pin$_3$	Pin$_4$	Pin$_5$	Pin$_6$	Pin$_7$	Pin$_8$	Pin$_9$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	0	0	0	1	0	1	0	1	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	1	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	1	0	0	0	1	0	0	1	0	0	1	0
3	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
4	1	1	1	0	1	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
5	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	1	1	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
7	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
8	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	1	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
9	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	0	1	0	1	1	1	1	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	0	1	0	0	1	0	0	1	0
10	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	0	0	0	1	0	1	0	0	1	0	1	0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
12	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	0	0	1	0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
13	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	1	0	1	1	1	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
14	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	0	0	0	1	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
15	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	1	0	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
16	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	1	1	0	0	0	0	1	0	1	0	0	1	0	1	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
17	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	1	0	0	1	0
18	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	1	0	0	1	0
19	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	0	1	1	1	0	0	1	0	1	0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
20	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	1	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0

Table B.17: FPGA outputs for Sample 1, Clk 34

FPGA	Pin_0	Pin_1	Pin_2	Pin_3	Pin_4	Pin_5	Pin_6	Pin_7	Pin_8	Pin_9	Pin_{10}	Pin_{11}	Pin_{12}	Pin_{13}	Pin_{14}	Pin_{15}	Pin_{16}	Pin_{17}	Pin_{18}	Pin_{19}	Pin_{20}	Pin_{21}	Pin_{22}	Pin_{23}	Pin_{24}	Pin_{25}	Pin_{26}	Pin_{27}	Pin_{28}	Pin_{29}	Pin_{30}	Pin_{31}	Pin_{32}	Pin_{33}	Pin_{34}	Pin_{35}	Pin_{36}	Pin_{37}	Pin_{38}	Pin_{39}	Pi_{n40}	Pin_{41}	Pin_{42}	Pin_{43}	Pin_{44}	Pin_{45}	Pin_{46}	Pin_{47}	Pin_{48}	Pin_{49}	Pin_{50}	Pin_{51}	Pin_{52}	Pin_{53}	Pin_{54}	Pin_{55}	Pin_{56}	Pin_{57}	Pin_{58}	Pin_{59}	Pin_{60}	Pin_{61}	Pin_{62}	Pin_{63}
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	1	1	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
3	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	1	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
4	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
5	1	1	1	0	1	1	0	1	1	1	0	0	1	0	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
7	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
8	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
9	1	1	1	0	1	1	0	1	1	1	0	0	1	0	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	0	1	0	0	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
10	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
12	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
13	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
14	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
15	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	1	0	1	1	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
16	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	1	1	0	1	0	0	0	1	0	0	1	0
18	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
19	1	1	1	0	1	1	0	1	1	1	1	0	1	0	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	1	1	0	1	0	1	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
20	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0

Table B.18: FPGA outputs for Sample 1, Clk 40

FPGA	Pin_0	Pin_1	Pin_2	Pin_3	Pin_4	Pin_5	Pin_6	Pin_7	Pin_8	Pin_9	Pin_{10}	Pin_{11}	Pin_{12}	Pin_{13}	Pin_{14}	Pin_{15}	Pin_{16}	Pin_{17}	Pin_{18}	Pin_{19}	Pin_{20}	Pin_{21}	Pin_{22}	Pin_{23}	Pin_{24}	Pin_{25}	Pin_{26}	Pin_{27}	Pin_{28}	Pin_{29}	Pin_{30}	Pin_{31}	Pin_{32}	Pin_{33}	Pin_{34}	Pin_{35}	Pin_{36}	Pin_{37}	Pin_{38}	Pin_{39}	Pi_{n40}	Pin_{41}	Pin_{42}	Pin_{43}	Pin_{44}	Pin_{45}	Pin_{46}	Pin_{47}	Pin_{48}	Pin_{49}	Pin_{50}	Pin_{51}	Pin_{52}	Pin_{53}	Pin_{54}	Pin_{55}	Pin_{56}	Pin_{57}	Pin_{58}	Pin_{59}	Pin_{60}	Pin_{61}	Pin_{62}	Pin_{63}
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	0	0	1	1	0	0	1	0	0	1	1	1	1	0	1	0	0	1	0	1	1	1	1	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	0	0	0	0	0	1	1	0	0	1	1	1	0	0	1	0	1	0	0	0	0	1	1	0	0	1	1	0	1	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0
3	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	0	0	0	0	0	1	1	1	0	1	1	1	0	1	0	0	0	0	1	0	0	1	1	0	1	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0
4	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	0	0	1	0	0	0	1	0	0	1	1	1	1	0	1	0	1	0	0	0	0	1	1	0	0	1	1	0	1	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0
5	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	0	0	0	1	0	0	1	0	0	1	1	1	1	0	1	0	1	1	0	0	0	1	1	0	0	1	1	0	1	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	0	1	1	0	0	1	1	0	0	1	1	1	1	0	1	0	1	0	0	0	0	1	1	0	0	1	1	0	1	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0
7	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	0	1	0	0	1	0	1	1	1	1	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0
8	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	1	0	0	0	1	0	1	1	1	0	1	1	1	0	0	1	0	1	0	0	0	0	1	1	0	0	1	1	0	1	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0
9	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	0	1	0	1	0	1	1	0	0	1	1	1	0	0	1	0	1	0	0	0	0	1	1	0	0	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
10	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	0	1	0	1	0	1	1	0	0	1	1	1	1	0	1	0	1	0	0	0	0	1	1	0	0	1	1	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	1	0	0	0	0	0	1	1	1	0	1	1	1	0	0	1	0	1	0	0	0	0	1	1	0	0	1	1	0	1	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0
12	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	1	0	0	1	0	0	0	1	0	1	1	1	0	1	1	1	0	0	1	0	1	0	0	0	0	1	1	0	0	1	1	0	1	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0
13	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	0	1	1	1	0	0	0	1	1	0	1	1	1	0	1	0	1	0	0	0	0	1	1	0	0	1	1	0	1	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0
14	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	0	0	1	1	0	0	1	0	0	1	1	1	1	0	1	0	0	1	0	1	1	1	1	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0
15	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	0	1	0	1	0	0	0	0	1	1	0	0	1	1	0	1	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0
16	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	0	1	0	1	1	1	1	0	0	1	1	0	0	0	1	0	1	0	0	0	0	1	1	0	0	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
17	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	0	0	0	1	0	0	1	0	0	1	1	1	1	0	1	0	1	0	0	0	0	1	1	0	0	1	1	0	1	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0
18	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	0	1	1	1	0	0	1	1	0	1	1	1	1	0	1	0	1	0	0	0	0	1	1	0	0	1	1	0	1	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0
19	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	0	1	0	1	1	0	0	0	1	1	0	0	1	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
20	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	0	1	0	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0

Table B.19: FPGA outputs for Sample 2, Clk 18

FPGA	Pin_0	Pin_1	Pin_2	Pin_3	Pin_4	Pin_5	Pin_6	Pin_7	Pin_8	Pin_9	Pin_{10}	Pin_{11}	Pin_{12}	Pin_{13}	Pin_{14}	Pin_{15}	Pin_{16}	Pin_{17}	Pin_{18}	Pin_{19}	Pin_{20}	Pin_{21}	Pin_{22}	Pin_{23}	Pin_{24}	Pin_{25}	Pin_{26}	Pin_{27}	Pin_{28}	Pin_{29}	Pin_{30}	Pin_{31}	Pin_{32}	Pin_{33}	Pin_{34}	Pin_{35}	Pin_{36}	Pin_{37}	Pin_{38}	Pin_{39}	Pi_{n40}	Pin_{41}	Pin_{42}	Pin_{43}	Pin_{44}	Pin_{45}	Pin_{46}	Pin_{47}	Pin_{48}	Pin_{49}	Pin_{50}	Pin_{51}	Pin_{52}	Pin_{53}	Pin_{54}	Pin_{55}	Pin_{56}	Pin_{57}	Pin_{58}	Pin_{59}	Pin_{60}	Pin_{61}	Pin_{62}	Pin_{63}
1	1	1	1	0	1	1	0	1	0	1	1	0	0	0	0	1	1	0	0	0	0	0	1	1	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	0	1	1	0	1	0	1	0	1	1	0	0	1	1	1	0	0	0	0	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	0	1	1	0	1	0	1	1	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	1	1	0	1	1	0	1	0	1	0	1	1	0	0	1	1	0	0	0	0	0	1	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0
5	1	1	1	0	1	1	0	1	0	1	1	0	1	0	0	1	1	1	0	1	0	0	1	0	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	1	1	0	1	1	0	1	0	1	1	1	0	0	0	1	1	1	0	0	0	0	1	1	0	1	0	1	1	0	1	0	0	1	1	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	1	1	0	1	1	0	1	0	1	0	1	1	0	0	1	1	1	0	0	0	0	1	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	0	1	0	0	1	0	1	0	0	1	0	0	1	1	1	0	0	0	0	1	1	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0
9	1	1	1	0	1	0	0	1	0	1	0	0	1	0	0	1	1	1	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0
10	1	1	1	0	1	1	0	1	0	1	0	1	0	0	0	1	1	1	0	0	0	0	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	1	1	0	1	0	0	1	0	1	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	1	1	1	0	1	0	0	1	0	1	0	1	1	0	0	1	1	1	0	1	0	0	1	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0
13	1	1	1	0	1	1	0	1	0	1	1	0	1	0	0	1	1	0	0	0	0	0	1	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	1	1	1	0	1	0	0	1	0	1	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0
15	1	1	1	0	1	1	0	1	0	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	1	1	1	0	1	0	0	1	0	1	0	0	1	0	0	1	1	1	0	0	0	0	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	1	1	1	0	0	1	0
17	1	1	1	0	0	0	0	1	0	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	1	1	0	0	0	0	1	0	1	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0
19	1	1	1	0	1	1	0	1	0	1	1	0	0	0	0	1	1	1	0	0	0	0	1	0	1	1	0	1	1	1	1	1	0	1	1	1	1	0	0	0	1	1	0	0	0	0	0	1	1	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0
20	1	1	1	0	1	1	0	1	0	1	1	0	0	0	0	1	1	0	0	0	0	0	1	1	0	1	0	0	1	0	1	0	0	1	1	1	1	0	0	0	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table B.20: FPGA outputs for Sample 2, Clk 19

FPGA	Pin_0	Pin_1	Pin_2	Pin_3	Pin_4	Pin_5	Pin_6	Pin_7	Pin_8	Pin_9	Pin_{10}	Pin_{11}	Pin_{12}	Pin_{13}	Pin_{14}	Pin_{15}	Pin_{16}	Pin_{17}	Pin_{18}	Pin_{19}	Pin_{20}	Pin_{21}	Pin_{22}	Pin_{23}	Pin_{24}	Pin_{25}	Pin_{26}	Pin_{27}	Pin_{28}	Pin_{29}	Pin_{30}	Pin_{31}	Pin_{32}	Pin_{33}	Pin_{34}	Pin_{35}	Pin_{36}	Pin_{37}	Pin_{38}	Pin_{39}	Pi_{n40}	Pin_{41}	Pin_{42}	Pin_{43}	Pin_{44}	Pin_{45}	Pin_{46}	Pin_{47}	Pin_{48}	Pin_{49}	Pin_{50}	Pin_{51}	Pin_{52}	Pin_{53}	Pin_{54}	Pin_{55}	Pin_{56}	Pin_{57}	Pin_{58}	Pin_{59}	Pin_{60}	Pin_{61}	Pin_{62}	Pin_{63}
1	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0
2	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	0	1	1	1	0	1	1	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0
3	1	1	1	0	1	0	0	1	1	1	0	0	1	0	1	0	0	1	1	1	0	1	1	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0
4	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1	1	1	0	1	1	1	1	0	0	0	1	0	1	0	1	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0
5	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	0	1	1	1	1	0	1	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0
6	1	1	1	0	1	0	0	1	1	1	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0
7	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	0	1	1	0	1	0	1	1	1	0	1	0	1	1	0	0	1	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
8	1	1	1	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0	1	1	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
9	1	1	1	0	1	0	0	1	1	1	0	0	1	0	1	0	0	1	1	0	1	0	1	1	1	0	1	0	0	1	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
10	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	1	0	1	1	0	1	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
11	1	1	1	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0	1	1	0	1	1	0	0	0	0	1	0	0	1	1	1	1	1	1	0	0	0	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
12	1	1	1	0	1	0	0	1	1	1	0	0	1	0	1	0	1	0	1	1	0	1	1	0	0	0	0	1	0	0	1	1	1	1	0	1	0	0	0	1	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
13	1	1	1	0	1	1	0	1	1	0	0	0	1	0	0	1	0	1	1	1	0	1	0	0	0	0	0	1	0	0	1	1	1	1	1	1	0	0	0	1	0	1	0	1	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0
14	1	1	1	0	1	0	0	1	0	1	0	0	1	0	1	0	0	1	1	1	0	1	0	0	0	0	0	1	0	0	1	1	1	1	1	1	0	0	0	1	0	1	0	1	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0
15	1	1	1	0	1	0	0	1	1	1	0	0	1	0	1	0	0	1	1	0	0	1	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0
16	1	1	1	0	1	0	0	1	1	1	0	0	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0	0	1	1	1	1	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
17	1	1	1	0	0	1	0	1	1	1	0	0	1	0	1	0	0	1	1	1	0	1	0	0	0	0	0	1	0	0	1	1	1	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0
18	1	1	1	0	0	0	0	1	0	1	0	0	1	0	1	0	0	1	1	1	0	1	1	0	0	0	0	1	0	0	1	1	1	1	1	1	0	0	0	1	0	1	0	1	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0
19	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	0	1	1	1	0	1	0	0	0	0	0	1	0	1	1	1	1	1	1	1	0	0	0	1	0	1	0	1	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0
20	1	1	1	0	1	1	0	1	1	0	0	0	1	0	0	1	0	1	1	0	0	1	1	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0

Table B.21: FPGA outputs for Sample 2, Clk 21

FPGA	Pin$_{0}$	Pin$_{1}$	Pin$_{2}$	Pin$_{3}$	Pin$_{4}$	Pin$_{5}$	Pin$_{6}$	Pin$_{7}$	Pin$_{8}$	Pin$_{9}$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	0	1	1	1	1	1	0	0	0	0	1	1
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	0	1	1	1	1	1	0	0	0	0	0	0
3	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	0	1	1	1	1	1	0	0	0	0	1	1
4	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	0	1	1	1	1	1	0	0	0	0	0	0
5	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	0	1	1	1	1	1	0	0	0	0	1	1
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	0	1	1	1	1	1	0	0	0	0	1	1
7	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
8	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
9	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	1	0	1	1	1	1	1	0	0	0	0	0	0
10	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1
12	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
13	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	0	1	1	1	1	1	0	0	0	0	1	1
14	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0
15	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	0	1	1	1	1	1	0	0	0	0	0	0
16	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	1	0	1	0	1	0	1	1	0	1	1	0	0	0	0	0	1
17	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0	0	0
18	1	1	1	0	0	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0
19	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	1	0	1	0	1	0	1	1	0	1	1	0	0	0	0	0	0
20	1	1	0	1	1	1	0	0	1	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	1	1	0	1	1	0	0	0

FPGA	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	0	0	1	1	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	0	0	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	1	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	1	1	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	1	0	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	1	0	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	1	1	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	1	1	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	1	0	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	1	0	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	1	0	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	1	1	0	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table B.22: FPGA outputs for Sample 2, Clk 24

FPGA	Pin$_{0}$	Pin$_{1}$	Pin$_{2}$	Pin$_{3}$	Pin$_{4}$	Pin$_{5}$	Pin$_{6}$	Pin$_{7}$	Pin$_{8}$	Pin$_{9}$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$
1	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	1	1	1	1	0	1	0	1	0	0	1	1	1	1	1
2	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	1	1	1	1	0	1	0	1	0	0	1	1	1	1	1
3	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	1	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1
4	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1
5	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	1	1	1	1	0	1	0	1	0	0	1	1	1	1	1
6	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	1	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1
7	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1	1	1	0
8	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	1	1	0	1	0	1	0	0	1	1	1	1	1
9	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1
10	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1
12	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0
13	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1	1	1	1
14	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	1	1	1	1	0	0	0	1	1	0	1	1	1	1	0
15	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	1	1	1	1	0	1	0	1	0	0	1	1	1	1	1
16	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	1	0	1	0	1	0	1	1	0	0	1	0	0	1	0
17	1	1	1	0	0	1	0	1	1	0	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	0	0	1	1	1	1	0
18	1	1	1	0	0	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	0	0	0	1	0	0	1	1	1	1	1
19	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0
20	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0	1	1	1	0

FPGA	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	1	1	1	1	0	0	0	1	1	0	1	0	1	1	1	1	0	1	0	1	0	1	1	1	1	1	1	0	0	0	1	0
2	1	1	0	0	1	0	1	1	0	0	0	0	1	0	1	0	0	1	0	1	0	1	1	1	1	1	1	0	0	0	1	0
3	1	1	0	0	1	0	1	1	0	0	0	0	1	0	1	0	0	1	0	0	1	0	1	1	1	1	1	0	0	0	1	0
4	1	1	1	1	0	0	0	1	1	0	1	0	1	1	1	1	0	1	0	1	0	1	1	1	1	1	1	0	0	0	1	0
5	1	1	0	0	1	1	1	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	1	1	1	1	0	0	0	1	0
6	1	1	1	1	0	0	0	1	1	0	1	0	1	1	1	1	0	1	0	0	0	1	1	1	1	1	1	0	0	0	1	0
7	1	1	0	0	1	0	1	1	0	0	0	0	1	0	1	0	1	0	0	0	1	0	1	1	1	1	1	0	0	0	1	0
8	1	1	0	0	1	0	1	1	0	0	0	0	1	0	1	0	1	0	0	1	0	1	1	1	1	1	1	0	0	0	1	0
9	0	1	1	1	0	0	0	1	1	0	1	0	1	1	1	0	1	0	0	0	1	0	1	1	1	1	1	0	0	0	1	0
10	1	1	0	0	1	0	1	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0	1	1	1	1	1	0	0	0	1	0
11	1	1	0	0	1	1	1	1	0	0	0	0	1	0	1	0	1	0	0	0	1	0	1	1	1	1	1	0	0	0	1	0
12	1	1	0	0	1	0	1	1	0	0	0	0	1	0	1	0	1	0	0	1	0	0	1	1	1	1	1	0	0	0	1	0
13	1	1	1	1	0	0	0	1	1	0	1	0	1	1	1	1	0	1	0	0	1	0	1	1	1	1	1	0	0	0	1	0
14	1	1	1	1	0	0	0	1	1	0	1	0	1	1	1	1	0	1	0	1	0	1	1	1	1	1	1	0	0	0	1	0
15	1	0	1	1	0	0	0	1	0	0	1	0	1	1	1	1	0	1	0	0	1	0	1	1	1	1	1	0	0	0	1	0
16	1	1	0	0	1	0	1	1	0	0	0	0	1	0	1	0	0	1	0	0	0	1	1	1	1	1	1	0	0	0	1	0
17	1	1	0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	1	0	1	0	0	1	1	1	1	1	0	0	0	1	0
18	1	1	1	1	0	0	0	1	1	1	1	0	1	1	1	1	0	1	0	0	0	0	1	1	1	1	1	0	0	0	1	0
19	1	1	0	0	1	0	1	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	1	1	1	1	0	0	0	1	0
20	1	1	1	1	1	0	0	1	1	1	1	0	1	1	1	1	0	1	0	1	0	1	1	1	1	1	1	0	0	0	1	0

Table B.23: FPGA outputs for Sample 2, Clk 25

FPGA	Pin$_0$	Pin$_1$	Pin$_2$	Pin$_3$	Pin$_4$	Pin$_5$	Pin$_6$	Pin$_7$	Pin$_8$	Pin$_9$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	1	0	1	1
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	1	0	0	1	0	1	1	0	1	0	0	0	1	0
3	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	1	0	0	1	0	1	1	0	1	0	0	0	1	0
4	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	1	0	1	1
5	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	0	1	1	0	0	0	1	1	1	1
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	1	1
7	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	1	0	1	1
8	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	0	1	1	1	0	0	1	1	1	1
9	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	0	1	1	0	0	0	1	0	1	1
10	1	1	1	0	1	1	0	1	0	1	0	0	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0	0	0	1	1	1
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0	1	1
12	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	0	1	1	0	0	0	1	0	1	1
13	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	0	1	1	1	0	0	1	1	1	1
14	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	0	0	1	1	0	1	1
15	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0	1	1
16	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	0	0	1	1	1	0	0	1	1	1
17	1	1	1	0	0	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	0	1	1	0	0	0	1	0	1	1
18	1	1	1	0	0	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	0	1	1	0	0	0	1	0	1	1
19	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	1	0	0	0	1	1	0	0	0	1	1	1	1	1	1	0
20	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	0	1	0	1	0	0	1	0	1	1

FPGA	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
2	1	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
3	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
4	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
5	1	0	1	0	1	0	1	0	0	1	0	1	1	1	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
6	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
7	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
8	0	1	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	1	0
9	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	1	0
10	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
11	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
12	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
13	1	0	1	1	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
14	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
15	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
16	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
17	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
18	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
19	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
20	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0

Table B.24: FPGA outputs for Sample 2, Clk 28

FPGA	Pin$_0$	Pin$_1$	Pin$_2$	Pin$_3$	Pin$_4$	Pin$_5$	Pin$_6$	Pin$_7$	Pin$_8$	Pin$_9$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	1	1	1	0	1	0	0	1	1	1
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	1	0	1	1	1	1	1	0	0	1	0	1
3	1	1	1	0	1	1	0	1	1	1	0	0	0	1	0	0	1	0	1	1	1	0	1	1	1	1	1	0	0	1	0	1
4	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	1	0	1	1	1	1	1	0	0	1	0	1
5	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	1	1	1	0	1	0	0	1	0	0	1	0
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	1	1	1	1	1	0	1	0	0	0	1	1
7	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	1	1	1	1	1	0	0	1	0	0	1	0
8	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	1	0	1	1	1	1	1	0	0	1	1	1
9	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	1	1	1	0	1	1	1	1	1	0	0	1	1	1
10	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	1	1	1	1	0	1	1	1	1	1	0	0	1	1	1
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1	1	1	0	0	1	1	1
12	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1	0	1	0	0	1	1	1
13	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	1	1	0	1	1	1	1	1	0	0	1	0	1
14	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	1	1	1	0	1	1	1	1	1	0	0	1	0	1
15	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	1	1	0	1	1	1	1	1	0	0	1	0	1
16	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	1	1	1	0	1	1	1	1	1	0	0	1	1	1
17	1	1	1	0	0	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	0	0	1	1	1	1	1	0	0	1	0	1
18	1	1	1	0	0	1	0	1	1	0	1	1	0	1	0	0	0	0	1	1	0	1	1	1	1	0	1	0	0	1	1	1
19	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1
20	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	1	0	1	1	1	1	0	1	0	0	1	1	1

FPGA	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	0	0	1	0	0	1	0	0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	0
2	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	0
3	0	1	1	0	0	1	0	0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	0
4	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0
5	0	1	0	0	0	1	0	0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	0
6	0	1	1	0	0	1	0	0	1	0	0	1	0	1	1	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0
7	1	0	1	0	0	1	0	0	1	0	0	0	0	1	1	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0
8	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
9	1	1	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
10	0	1	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0
11	1	0	1	0	0	1	0	0	1	0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	0
12	1	0	1	0	0	1	0	0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	0
13	0	1	1	0	0	1	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	0
14	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	0
15	0	1	1	0	0	1	0	0	1	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
17	0	0	0	0	0	1	0	0	1	0	0	1	0	1	1	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0
18	0	1	1	0	0	1	0	0	1	0	0	1	0	1	1	1	1	1	1	1	0	1	1	1	0	0	0	1	0	0	1	0
19	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
20	0	0	0	0	0	1	0	0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	0

Table B.25: FPGA outputs for Sample 2, Clk 30

FPGA	Pin$_0$	Pin$_1$	Pin$_2$	Pin$_3$	Pin$_4$	Pin$_5$	Pin$_6$	Pin$_7$	Pin$_8$	Pin$_9$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$
1	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	0	1	0	1	1	1	1	0	0	0	0	1	0	0	0	0
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	1	0	0	0	1	1	0	0	0	0	0	1	1	1	0
3	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	0	1	0	1	1	1	1	0	0	0	0	1	0	0	0	0
4	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0	1	0	0	0	0
5	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	0	0	0	0
6	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	0	1	0	0	0	1	1	0	0	0	0	0	0	1	1	0
7	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	1	0	0	1	1	1	1	1	0	0	0	1	0	0	0	1
8	1	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	1	0	0	1	1	1	1	0	1	0	0	1	0	0	0	0
9	1	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	1	1	1	0
10	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	1	0	0	1	1	1	1	0	0	0	0	1	0	0	0	0
11	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	1	0	0	1	1	1	1	0	0	0	0	1	0	0	0	0
12	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	1	0	0	1	1	1	1	1	0	0	0	1	0	0	0	0
13	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	0	1	0	0	0	1	1	0	0	0	0	0	1	1	1	0
14	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	0	1	0	1	1	1	1	0	0	0	0	1	0	0	0	0
15	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	0	1	0	1	1	1	1	0	0	0	0	1	0	1	0	0
16	1	1	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	1	0
17	1	1	1	0	0	1	0	1	1	0	1	1	0	0	1	1	0	1	0	1	0	1	1	0	0	0	0	1	0	0	0	0
18	1	1	1	0	0	1	0	1	1	0	1	0	0	0	0	1	0	1	0	0	0	1	1	0	0	0	0	0	1	1	1	0
19	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	1	0	1	1	1	0	0	0	1	1
20	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	1

FPGA	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
2	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
3	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
4	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
5	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
6	1	0	0	0	1	1	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
7	0	1	0	0	0	1	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
8	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0	1	1	0	0	0	1	0	0	1	0	0	1	0
9	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
10	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
11	0	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0
12	0	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	0	0	1	0	0	1	0
13	1	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0
14	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1	0	0	1	0	0	1	0
15	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
16	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
17	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	1	1	1	1	0	1	1	1	0	0	0	0	0	0	1	0
18	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	1	1	1	1	0	1	1	1	0	0	0	0	0	0	1	0
19	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
20	1	0	1	0	1	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0

Table B.26: FPGA outputs for Sample 2, Clk 34

FPGA	Pin$_0$	Pin$_1$	Pin$_2$	Pin$_3$	Pin$_4$	Pin$_5$	Pin$_6$	Pin$_7$	Pin$_8$	Pin$_9$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$
1	1	1	1	0	1	1	0	1	1	1	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1
2	1	1	1	0	1	1	0	1	1	1	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1
3	1	1	1	0	1	1	0	1	1	1	0	0	1	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
4	1	1	1	0	1	1	0	1	1	1	0	0	1	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1
5	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0
6	1	1	1	0	1	1	0	1	1	1	0	0	1	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
7	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	1	0
8	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	1	1
9	1	1	1	0	0	0	1	1	0	0	0	1	1	1	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	1	1
10	1	1	1	0	1	1	0	1	1	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	0	1	0
11	1	1	1	0	1	1	0	1	1	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	0	1	1
12	1	1	1	0	1	1	0	1	1	1	1	1	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	0	1	0
13	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1
14	1	1	1	0	1	1	0	1	1	1	1	1	0	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1
15	1	1	1	0	1	1	0	1	1	1	1	0	0	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1
16	1	1	1	0	1	1	0	1	1	1	1	1	0	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0
17	1	1	1	0	0	1	0	1	1	1	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
18	1	1	1	0	0	1	0	1	1	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
19	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1
20	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	0	1	1	0	1	1	0	0	1	0	1	0	1	1

FPGA	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	1	0	1	1	0	1	0	0	0	1	0	0	1	0
2	1	1	0	1	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
3	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
4	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	1	0	1	1	0	1	0	0	0	1	0	0	1	0
5	1	0	1	1	1	1	1	1	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	1	0	0	1	1	1	1	0	0	0	0	1	0	0	1	0	0	1	1	0	1	0	1	0	0	0	1	0	0	1	0
7	1	1	0	1	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	1	1	0	1	0	0	0	1	0	0	1	0
8	1	1	0	0	1	1	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
10	1	1	0	1	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
12	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	1	0	0	0	1	0	0	1	0
13	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	1	1	0	1	0	1	0	0	0	1	0	0	1	0
14	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	1	0	1	1	1	1	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
16	1	0	1	0	1	1	1	1	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	1	0	1	1	1	0	0	0	1	0	0	1	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	1	1	0	1	0	1	0	0	0	1	1	0	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0

Table B.27: FPGA outputs for Sample 2, Clk 40

FPGA	Pin$_0$	Pin$_1$	Pin$_2$	Pin$_3$	Pin$_4$	Pin$_5$	Pin$_6$	Pin$_7$	Pin$_8$	Pin$_9$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$
1	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	1	1	0	0	0	0	0	1	1	1	0	1	1	1	0
2	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0
3	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0
4	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	1	0	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0
5	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	1	0	1	1	0	0	1	0	0	1	1	0	0	1	1	1	0
6	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	1	0	0	0	0	0	1	1	1	0	1	1	1	0
7	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0
8	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0
9	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0
10	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0
11	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0
12	1	1	1	0	1	1	0	0	0	1	0	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0
13	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	1	0	1	1	0	0	0	1	1	1	1	1	0	1	1	1	0
14	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	1	1	0	0	0	1	0	1	1	0	0	1	1	1	1
15	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0
16	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0
17	1	1	1	0	0	1	0	1	1	0	1	1	0	0	1	1	0	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0
18	1	1	1	0	0	1	0	1	1	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0	1	1	1	0	1	1	1	0
19	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0
20	1	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1	0	1	1	1	0

FPGA	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	0	1	0	1	0	0	0	0	1	1	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
3	0	1	0	0	1	0	1	1	1	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
6	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
7	1	0	0	1	1	0	0	0	1	1	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1	0	0	1	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
14	0	1	0	1	0	0	0	0	1	1	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	1	1	1	1	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	1	1	1	1	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
20	0	1	0	0	1	0	1	1	1	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0

Table B.28: FPGA outputs for Sample 3, Clk 18

FPGA	Pin$_0$	Pin$_1$	Pin$_2$	Pin$_3$	Pin$_4$	Pin$_5$	Pin$_6$	Pin$_7$	Pin$_8$	Pin$_9$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$
1	1	1	1	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	0	0	0	0	1	0	0	1	0	0	1	1	0	1
2	1	1	1	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	0	1	0	0	0	0	0	1	1	0	1
3	1	1	1	0	0	0	0	1	0	1	0	1	0	1	1	1	1	1	0	1	0	1	1	0	0	1	0	0	1	1	0	1
4	1	1	1	0	0	0	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	0	1	0	0	1	0	0	1	1	0	1
5	1	1	1	0	0	0	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	1	1	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	0	1	0	0	0	0	0	1	1	0	1
7	1	1	1	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	0	1	0	0	0	0	0	1	1	0	1
8	1	1	1	0	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	1	0	1	1	0	0	0	0	0	1	1	0	1
9	1	1	1	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1	1	0	0	0	0	0	1	1	0	1
10	1	1	1	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1	1	0	0	1	0	0	1	1	0	1
11	1	1	1	0	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	1	0	0	1	0	0	1	0	0	1	1	0	1
12	1	1	1	0	0	0	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	1	1	1	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	0	1	0	0	0	0	0	1	1	0	1
14	1	1	1	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1	1	0	0	0	0	0	1	1	0	1
15	1	1	1	0	0	0	0	1	0	1	0	1	0	1	1	1	1	1	0	1	0	0	1	0	0	0	0	0	1	1	0	1
16	1	1	1	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1	1	0	0	0	0	0	1	1	0	1
17	1	1	1	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	0	1	0	0	1	0	0	1	1	0	1
18	1	1	1	0	0	0	0	1	0	0	0	1	0	1	1	1	1	1	0	1	0	1	1	0	0	0	0	0	1	1	0	1
19	1	1	1	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1	1	0	0	1	0	0	1	1	0	1
20	1	1	1	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	0	0	0	0	1	0	0	0	0	0	1	1	0	1

FPGA	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table B.29: FPGA outputs for Sample 3, Clk 19

FPGA	Pin$_{0}$	Pin$_{1}$	Pin$_{2}$	Pin$_{3}$	Pin$_{4}$	Pin$_{5}$	Pin$_{6}$	Pin$_{7}$	Pin$_{8}$	Pin$_{9}$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1	1	0	1	0	0	1	1	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1	1	0	1	1	1	1	1	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1	1	0	1	0	1	1	1	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1	1	0	1	0	0	1	1	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1	1	0	1	0	0	1	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1	1	0	1	0	1	1	0	0
13	1	1	0	0	1	1	0	0	1	1	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1	1	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	0	0	0	1	0	1	1	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1	1	0	1	0	0	1	1	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	1	1	1	1	1
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1	1	0	1	0	1	1	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0
19	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	0	1	1	0
20	1	1	0	0	1	0	0	0	1	1	0	0	1	0	1	0	1	0	1	0	1	1	1	1	0	0	0	0	1	1	1	0

FPGA	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0	1	1	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0	1	1	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	0	0

Table B.30: FPGA outputs for Sample 3, Clk 21

FPGA	Pin$_{0}$	Pin$_{1}$	Pin$_{2}$	Pin$_{3}$	Pin$_{4}$	Pin$_{5}$	Pin$_{6}$	Pin$_{7}$	Pin$_{8}$	Pin$_{9}$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$
1	1	1	0	1	1	1	0	0	1	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0	0	0
2	1	1	0	1	1	1	0	1	1	0	1	1	0	0	0	1	1	1	0	1	0	0	1	0	1	1	0	1	1	0	0	0
3	1	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	1	1	0	0	0
4	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	0	1	0	0	1	0	1	1	0	1	1	0	0	0
5	1	1	0	1	1	1	0	1	1	0	1	1	0	0	0	1	0	1	0	1	0	0	1	0	0	1	0	1	1	0	0	0
6	1	1	0	1	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	1	1	0	0	0
7	1	1	0	1	1	1	0	0	1	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0	0	0
8	1	1	0	1	1	1	0	1	1	0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1	1	0	1	1	0	0	0
9	1	1	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	1	0	1	1	0	0	0
10	1	1	0	1	1	1	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	1	1	0	1	1	0	0	0
11	1	1	0	1	1	1	0	1	1	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0	0	0
12	1	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0	0	0
13	1	1	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	1	0	1	1	0	0	0
14	1	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0	0	0
15	1	1	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	1	0	1	1	0	0	0
16	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	1	0	1	1	0	0	0
17	1	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0	0	0
19	1	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	1	1	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FPGA	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	1	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	1	1	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	1	1	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	1	0	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	1	1	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	1	1	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	1	0	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table B.31: FPGA outputs for Sample 3, Clk 24

FPGA	Pin$_{0}$	Pin$_{1}$	Pin$_{2}$	Pin$_{3}$	Pin$_{4}$	Pin$_{5}$	Pin$_{6}$	Pin$_{7}$	Pin$_{8}$	Pin$_{9}$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$
1	1	1	1	0	1	1	0	1	1	0	0	1	0	1	0	0	0	1	0	1	0	1	1	1	1	0	0	1	1	0	1	0
2	1	1	1	0	1	1	0	1	1	0	0	1	0	1	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0	0	1	0
3	1	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	1	0	1	0	1	1	1	1	0	0	1	1	0	1	0
4	1	1	1	0	1	1	0	1	1	0	0	1	0	1	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0	0	1	0
5	1	1	1	0	1	1	0	1	1	0	0	1	0	1	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0	0	1	0
6	1	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	0	1	0	1	0	1	1	1	1	0	0	1	1	0	1	0
7	1	1	1	0	1	1	0	1	1	0	0	1	0	1	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0	0	1	0
8	1	1	1	0	1	1	0	1	1	0	0	1	0	1	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0	0	1	0
9	1	1	1	0	1	1	0	1	1	0	0	1	1	1	0	0	0	1	0	1	0	1	0	1	1	0	0	1	0	0	1	0
10	1	1	1	0	1	1	0	1	1	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	1	1	0
11	1	1	1	0	1	1	0	1	1	0	0	1	0	1	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0	0	1	0
12	1	1	1	0	1	1	0	1	1	0	0	1	0	1	0	0	1	1	1	1	1	0	1	0	1	0	0	1	1	1	1	1
13	1	1	1	0	1	1	0	1	1	0	0	1	0	1	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0	0	1	0
14	1	1	1	0	1	1	0	1	1	0	0	1	0	1	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0	0	1	0
15	1	1	1	0	1	1	0	1	1	0	0	1	0	1	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0	0	1	0
16	1	1	1	0	1	1	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	1	1	1	0	1	1	0	1	1	0	0	1	0	1	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0	0	1	0
18	1	1	1	0	1	1	0	1	1	0	0	1	0	1	0	0	0	1	0	1	0	1	1	1	1	0	0	1	1	1	1	0
19	1	1	1	0	0	1	0	1	0	0	0	0	1	1	0	1	1	1	1	1	1	0	0	0	1	0	0	1	1	1	1	0
20	1	1	1	0	1	1	0	1	1	0	0	1	0	0	0	0	1	1	1	1	1	0	0	0	1	1	0	1	1	1	1	0

FPGA	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	1	0	1	1	0	0	0	0	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	1	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	1	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	1	1	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	1	1	1	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	0	1	1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	1	1	1	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	0	1	1	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	1	1	1	1	0	0	0	1	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	1	1	1	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	1	1	1	1	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	1	0	1	1	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	1	0	1	1	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	1	0	1	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	1	0	1	1	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	1	1	1	1	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	0	1	1	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	1	0	1	1	0	0	0	1	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	1	0	1	1	1	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table B.32: FPGA outputs for Sample 3, Clk 25

FPGA	Pin$_{0}$	Pin$_{1}$	Pin$_{2}$	Pin$_{3}$	Pin$_{4}$	Pin$_{5}$	Pin$_{6}$	Pin$_{7}$	Pin$_{8}$	Pin$_{9}$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$
1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	1	0	1	0	1	0	0	1	1	1	1
3	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	1	0	1	0	1	0	1	1	0	1	1
4	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	1	1	0	1	0	1	1	0	0	0	1	0	1	1
17	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	1	1	1	0	1	1	0	1	0	1	0	0	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0	0	1	1	1	1
20	1	1	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FPGA	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	1	0	1	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
2	1	0	1	0	1	0	1	0	0	0	0	1	1	1	1	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
3	1	0	0	0	1	0	1	0	0	0	0	1	1	1	1	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
4	1	0	1	0	1	0	1	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
5	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
6	0	0	1	0	1	0	1	0	0	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
7	1	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
8	1	0	1	0	1	0	1	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	1	0	0	1	0	1	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	0	1	0	1	0	1	0	0	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
11	1	0	1	0	1	0	1	0	0	1	0	1	1	1	1	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
12	1	0	0	0	1	0	1	0	0	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
13	1	0	0	0	1	0	1	0	0	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
14	1	0	0	0	1	0	1	0	0	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
15	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
16	1	0	1	0	1	0	1	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	1	0	1	0	1	0	1	0	0	0	0	1	1	1	1	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
18	1	0	0	0	0	0	0	1	0	1	0	1	0	0	1	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
19	0	0	0	0	1	0	1	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	1	0	0	1	0

Table B.33: FPGA outputs for Sample 3, Clk 28

FPGA	Pin$_{0}$	Pin$_{1}$	Pin$_{2}$	Pin$_{3}$	Pin$_{4}$	Pin$_{5}$	Pin$_{6}$	Pin$_{7}$	Pin$_{8}$	Pin$_{9}$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$
1	1	1	1	0	1	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1
2	1	1	1	0	1	1	0	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	1	1	1	1	0	0	1	1
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1
4	1	1	1	0	1	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1
5	1	1	1	0	1	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1
6	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1
7	1	1	1	0	1	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1
8	1	1	1	0	1	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1
9	1	1	1	0	1	1	0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	1
10	1	1	1	0	1	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1
11	1	1	1	0	1	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1
12	1	1	1	0	1	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1
13	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1
14	1	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1
15	1	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1
16	1	1	1	0	1	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1
17	1	1	1	0	1	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1
18	1	1	1	0	1	1	0	1	1	1	0	0	0	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1
19	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1
20	1	1	1	0	1	1	0	1	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	1	1	1	0	0	1	1

FPGA	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	1	0	0	1	0	0	1	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	1	0	0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	0
8	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	0
9	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
10	0	1	1	0	0	1	0	0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	1	1	0	0	1	0	0	1	0	0	0	0	1	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	1	1	0	0	1	0	0	1	0	0	1	0	1	1	1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table B.34: FPGA outputs for Sample 3, Clk 30

FPGA	Pin$_{0}$	Pin$_{1}$	Pin$_{2}$	Pin$_{3}$	Pin$_{4}$	Pin$_{5}$	Pin$_{6}$	Pin$_{7}$	Pin$_{8}$	Pin$_{9}$	Pin$_{10}$	Pin$_{11}$	Pin$_{12}$	Pin$_{13}$	Pin$_{14}$	Pin$_{15}$	Pin$_{16}$	Pin$_{17}$	Pin$_{18}$	Pin$_{19}$	Pin$_{20}$	Pin$_{21}$	Pin$_{22}$	Pin$_{23}$	Pin$_{24}$	Pin$_{25}$	Pin$_{26}$	Pin$_{27}$	Pin$_{28}$	Pin$_{29}$	Pin$_{30}$	Pin$_{31}$
1	1	1	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	1	1	1	0
2	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	0	1	1	1	0	0	0	0	1	0	0	0	1	1	1	1
3	1	1	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	1	0
4	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	0	1	0	0	0	1	1	0	0	0	0	0	1	1	1	0
5	1	1	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	1	1	0
6	1	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	1	1	1	1
7	1	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	1	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	1	1	1	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	1	1	1	1
10	1	1	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	1	1	1	0
11	1	1	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	1	0
12	1	1	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	1	1	1	0
13	1	1	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	1	1	1	1
14	1	1	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	1	1	1	0
15	1	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	1	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	1	0	0	0	0
17	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	0	1	0	0	0	1	1	0	0	0	0	0	1	1	1	0
18	1	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	1	1	1	1
19	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	0	1	0	1	1	1	1	1	0	0	0	1	0	0	0	0
20	1	1	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	1	1	1	1	0	0	1	0	1	0	1	0	0

FPGA	Pin$_{32}$	Pin$_{33}$	Pin$_{34}$	Pin$_{35}$	Pin$_{36}$	Pin$_{37}$	Pin$_{38}$	Pin$_{39}$	Pi$_{n40}$	Pin$_{41}$	Pin$_{42}$	Pin$_{43}$	Pin$_{44}$	Pin$_{45}$	Pin$_{46}$	Pin$_{47}$	Pin$_{48}$	Pin$_{49}$	Pin$_{50}$	Pin$_{51}$	Pin$_{52}$	Pin$_{53}$	Pin$_{54}$	Pin$_{55}$	Pin$_{56}$	Pin$_{57}$	Pin$_{58}$	Pin$_{59}$	Pin$_{60}$	Pin$_{61}$	Pin$_{62}$	Pin$_{63}$
1	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
2	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	1	1	0	1	0	1	0	1	1	0	0	0	1	0
3	1	0	0	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
4	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1	0	0	1	0	0	1	0
5	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1	0	0	1	0	0	1	0
6	0	1	1	1	0	0	0	1	1	1	1	1	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0
7	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1	0	0	1	0	0	1	0
8	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	1	0
10	1	0	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	1	0	0	1	0
11	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
12	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
13	1	0	0	0	1	0	1	0	1	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
14	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0
15	1	0	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0
16	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	1	0	0	1	0
17	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	1	0
18	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	1	1	0	0	0	1	0
19	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1	0	0	1	0	0	1	0
20	1	1	1	1	0	0	0	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0

Table B.35: FPGA outputs for Sample 3, Clk 34

FPGA	Pin_{0}	Pin_{1}	Pin_{2}	Pin_{3}	Pin_{4}	Pin_{5}	Pin_{6}	Pin_{7}	Pin_{8}	Pin_{9}	Pin_{10}	Pin_{11}	Pin_{12}	Pin_{13}	Pin_{14}	Pin_{15}	Pin_{16}	Pin_{17}	Pin_{18}	Pin_{19}	Pin_{20}	Pin_{21}	Pin_{22}	Pin_{23}	Pin_{24}	Pin_{25}	Pin_{26}	Pin_{27}	Pin_{28}	Pin_{29}	Pin_{30}	Pin_{31}	Pin_{32}	Pin_{33}	Pin_{34}	Pin_{35}	Pin_{36}	Pin_{37}	Pin_{38}	Pin_{39}	Pi_{n40}	Pin_{41}	Pin_{42}	Pin_{43}	Pin_{44}	Pin_{45}	Pin_{46}	Pin_{47}	Pin_{48}	Pin_{49}	Pin_{50}	Pin_{51}	Pin_{52}	Pin_{53}	Pin_{54}	Pin_{55}	Pin_{56}	Pin_{57}	Pin_{58}	Pin_{59}	Pin_{60}	Pin_{61}	Pin_{62}	Pin_{63}
1	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	1	1	0	1	0	1	0	0	0	1	0	0	1	0
3	1	1	1	0	0	0	1	1	0	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1	0	1	1	1	1	0	1	0	1	0	0	1	1	1	1	0	0	1	0	1	1	0	1	0	0	0	1	0	0	1	0
4	1	1	1	0	0	0	1	1	0	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	1	1	0	0	0	1	1	0	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	1	1	0	1	1	0	1	1	1	1	0	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	0	1	0	1	1	0	1	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	0	1	1	0	1	1	1	0	1	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	1	0	0	1	0
10	1	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1	0	1	0	1	1	1	1	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	1	1	0	1	0	1	0	0	0	1	0	0	1	0
12	1	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	1	1	1	0	1	1	0	1	1	1	0	0	0	0	0	1	1	1	1	0	1	1	0	1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	1	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	1	1	1	0	1	1	0	1	1	1	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	1	1	0	0	1	0	1	1	0	1	0	1	1	0	1	0	1	0	1	1	1	1	1	0	0	1	0	1	1	0	1	0	0	0	1	0	0	1	0

Table B.36: FPGA outputs for Sample 3, Clk 40

FPGA	Pin_{0}	Pin_{1}	Pin_{2}	Pin_{3}	Pin_{4}	Pin_{5}	Pin_{6}	Pin_{7}	Pin_{8}	Pin_{9}	Pin_{10}	Pin_{11}	Pin_{12}	Pin_{13}	Pin_{14}	Pin_{15}	Pin_{16}	Pin_{17}	Pin_{18}	Pin_{19}	Pin_{20}	Pin_{21}	Pin_{22}	Pin_{23}	Pin_{24}	Pin_{25}	Pin_{26}	Pin_{27}	Pin_{28}	Pin_{29}	Pin_{30}	Pin_{31}	Pin_{32}	Pin_{33}	Pin_{34}	Pin_{35}	Pin_{36}	Pin_{37}	Pin_{38}	Pin_{39}	Pi_{n40}	Pin_{41}	Pin_{42}	Pin_{43}	Pin_{44}	Pin_{45}	Pin_{46}	Pin_{47}	Pin_{48}	Pin_{49}	Pin_{50}	Pin_{51}	Pin_{52}	Pin_{53}	Pin_{54}	Pin_{55}	Pin_{56}	Pin_{57}	Pin_{58}	Pin_{59}	Pin_{60}	Pin_{61}	Pin_{62}	Pin_{63}
1	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	1	1	0	1	0	1	0	0	0	1	0	0	1	0
3	1	1	1	0	0	0	1	1	0	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1	0	1	1	1	1	0	1	0	1	0	0	1	1	1	1	0	0	1	0	1	1	0	1	0	0	0	1	0	0	1	0
4	1	1	1	0	0	0	1	1	0	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	1	1	0	0	0	1	1	0	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	1	1	0	1	1	0	1	1	1	1	0	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	0	1	0	1	1	0	1	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	0	1	1	0	1	1	1	0	1	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	1	0	0	1	0
10	1	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1	0	1	0	1	1	1	1	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	1	1	0	1	0	1	0	0	0	1	0	0	1	0
12	1	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	1	1	1	0	1	1	0	1	1	1	0	0	0	0	0	1	1	1	1	0	1	1	0	1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	1	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	1	0	0	0	0	1	1	1	0	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	1	1	1	0	1	1	0	1	1	1	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	1	1	0	0	1	0	1	1	0	1	0	1	1	0	1	0	1	0	1	1	1	1	1	0	0	1	0	1	1	0	1	0	0	0	1	0	0	1	0

Listing C.1: Nodal Cumulative Sampling Circuit Entities

```
1  ENTITY glitch_capture_setup IS
2          PORT(set : IN STD_LOGIC;
3                  reset : IN STD_LOGIC;
4                  clock_lfsr : IN STD_LOGIC;
5                  clock_reg : IN STD_LOGIC;
6                  pin_out : OUT STD_LOGIC;
7                  reg_values : OUT STD_LOGIC_VECTOR(11 DOWNTO 0));
8  END ENTITY;
9
10 ENTITY glitch_capture IS
11         GENERIC(mult_width : POSITIVE :=4);
12         PORT(x : IN STD_LOGIC_VECTOR(mult_width-1 DOWNTO 0);
13                 y : IN STD_LOGIC_VECTOR(mult_width-1 DOWNTO 0);
14                 reset : IN STD_LOGIC;
15                 pin_out : OUT STD_LOGIC;
16                 output_vals : OUT STD_LOGIC_VECTOR((mult_width)-1 ...
                      DOWNTO 0));  --vector of multiplier outputs
17 END ENTITY;
18
19 ENTITY lfsr IS
20         GENERIC (width : positive);
21         PORT(   clock : in std_logic;
22                 load : in std_logic_vector( width - 1 DOWNTO 0);
23                 bit_out : out std_logic;
24                 state_out : out std_logic_vector (width - 1 DOWNTO...
                       0));
25 END ENTITY lfsr;
26
27 ENTITY multip_cell IS
28     PORT ( p_in : IN std_logic;
29               x : IN std_logic;
30               y : IN std_logic;
31               c_in : IN std_logic;
32               p_out : OUT std_logic;
33               c_out : OUT std_logic);
34 END ENTITY multip_cell;
35
36 ENTITY multiplier IS
37     GENERIC( width : positive :=4);
38     PORT(x : in std_logic_vector(width -1 downto 0);
39            y : in std_logic_vector(width -1 downto 0);
40            product : out std_logic_vector((2*width) -1 downto 0));
41 END ENTITY;
42
43 ENTITY storage IS
44     GENERIC(size : POSITIVE :=4);
45     PORT(clock : IN STD_LOGIC;
46         reset : IN STD_LOGIC; --active low
47         input : IN STD_LOGIC_VECTOR(size-1 DOWNTO 0);
48         output :OUT STD_LOGIC_VECTOR(size-1 DOWNTO 0));
```

```
49 END ENTITY storage;
```

Listing C.2: Transitional Sampling Circuit Entities

```
1  ENTITY glitch_read_setup IS
2          PORT(set : IN STD_LOGIC;
3                  reset : IN STD_LOGIC;
4                  clock : IN STD_LOGIC;
5                  read_bits0 : OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
6                  read_bits1 : OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
7                  read_bits2 : OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
8                  read_bits3 : OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
9                  read_bits4 : OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
10                 read_bits5 : OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
11                 read_bits6 : OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
12                 read_bits7 : OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
13                 read_bits8 : OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
14                 read_bits9 : OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
15                 read_bits10 : OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
16                 read_bits11 : OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
17                 read_bits12 : OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
18                 read_bits13 : OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
19                 read_bits14 : OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
20                 read_bits15 : OUT STD_LOGIC_VECTOR(15 DOWNTO 0));
21 END ENTITY;
22
23 ENTITY glitch_read IS
24         GENERIC(mult_width : POSITIVE :=4);
25         PORT(x : IN STD_LOGIC_VECTOR(mult_width-1 DOWNTO 0);
26                 y : IN STD_LOGIC_VECTOR(mult_width-1 DOWNTO 0);
27                 reset : IN STD_LOGIC;
28                 sample0 : OUT STD_LOGIC_VECTOR((2*mult_width) -1 ...
                       DOWNTO 0);
29                 sample1 : OUT STD_LOGIC_VECTOR((2*mult_width) -1 ...
                       DOWNTO 0);
30                 sample2 : OUT STD_LOGIC_VECTOR((2*mult_width) -1 ...
                       DOWNTO 0);
31                 sample3 : OUT STD_LOGIC_VECTOR((2*mult_width) -1 ...
                       DOWNTO 0));
32 END ENTITY;
33
34 ENTITY lfsr IS
35         GENERIC (width : positive);
36         PORT(    clock : in std_logic;
37                  load : in std_logic_vector( width - 1 DOWNTO 0);
38                  bit_out : out std_logic;
39                  state_out : out std_logic_vector (width - 1 DOWNTO...
                        0));
40 END ENTITY lfsr;
41
42 ENTITY multip_cell IS
43     PORT ( p_in : IN std_logic;
44              x : IN std_logic;
```

```
45              y : IN std_logic;
46              c_in : IN std_logic;
47              p_out : OUT std_logic;
48              c_out : OUT std_logic);
49 END ENTITY multip_cell;
50
51 ENTITY multiplier IS
52     GENERIC( width : positive :=4);
53     PORT(x : in std_logic_vector(width -1 downto 0);
54            y : in std_logic_vector(width -1 downto 0);
55            product : out std_logic_vector((2*width) -1 downto 0));
56 END ENTITY;
57
58 ENTITY shifter_sample IS
59         PORT(reset : IN STD_LOGIC;
60            set : IN STD_LOGIC;
61                 clock : IN STD_LOGIC;
62                 d : IN STD_LOGIC;
63                 s0_out : OUT STD_LOGIC;
64                 s1_out : OUT STD_LOGIC;
65                 s2_out : OUT STD_LOGIC;
66                 s3_out : OUT STD_LOGIC);
67 END ENTITY;
```

Bibliography

1. Abramovici, Miron, Melvin A. Breuer, and Arthur D. Friedman. *Digital Systems Testing and Testable Design*. IEEE Press, 1990. ISBN 0-7803-1062-4.

2. Ashenden, Peter J. *The Designer's Guide to VHDL*. Morgan Kaufmann, second edition, 2002. ISBN 978-1-55860-674-6.

3. Bar-El, Hagai. *Introduction to Side Channel Attacks*. White paper, Discretix Technologies Ltd.

4. Bernstein, Kerry. *High Speed CMOS Design Styles*. Kluwer Academic Publishers, 1998.

5. Boning, Duane and Sani Nassif. *Design of High Performance Microprocessor Circuits*, chapter Models of Process Variations in Device and Interconnect. Wiley-IEEE Press, 2000.

6. Bushnell, Michael L. and Vishwani D. Agrawal. *Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits*. Springer Science and Business Media, Inc., 2000. ISBN 0792379918.

7. Corporation, Intel. "Intel Product Comparison Chart", Nov 2007.

8. Defense, Department of. *The Defense Acquisition System*. Washington: GPO, May 2003.

9. Dipert, Brian. "Cunning Circuits Confound Crooks". October 2000.

10. Franssila, Sami. *Introduction to Microfabrication*. Wiley, 2004. ISBN 0-470-85105-8.

11. Gassend, Blaise, Dwaine Clarke, Marten van Dijk, and Srinivas Devadas. "Silicon Physical Random Functions". *CCs '02: Proceedings of the 9th ACM Conference on Computer and Communications Security*, 148–160. ACM, 2002.

12. Goldberg, Ian and David Wagner. *Cracking DES: Secrets of Encryption Research, Wiretap Politics and Chip Design*, chapter 10 Architectural Considerations for Cryptanalytic Hardware. O'Reilly Media, 1998.

13. Homeland Security, Department of. "Two Charged and Arrested in Scheme to Illegally Export U.S. Fighter Jet Components to China". Webpage, July 2003.

14. Kocher, Paul, Joshua Jaffe, and Benjamin Jun. "Differential Power Analysis". *Lecture Notes in Computer Science*, 1666:388–397, 1999.

15. Leppvuori, Seppo. "Electronics Materials, Packaging and Reliability Techniques (EMPART)". Webpage, February 2008. URL `www.infotech.oulu.fi/Annual/2000/EMPART.html`.

16. Samyde, David, Sergei Skorobogatov, Ross Anderson, and Jean-Jacques Quisuater. "On a New Way to Read Data from Memory". *SISW '02: Proceedings of the First International IEEE Security in Storage Workshop*, 65. IEEE Computer Society, Washington, DC, USA, 2002. ISBN 0-7695-1888-5.

17. Skorobogatov, Sergei P. and Ross J. Anderson. "Optical fault induction attacks", 2002.

18. Suh, G. Edward and Srinivas Devadas. "Physical Uncloneable Functions for Device Authentication and Secret Key Generation". *DAC '07: Proceedings of the 44th Annual Conference on Design Automation*, 9–14. ACM, 2007.

19. Weste, Neil H. E. and David Harris. *CMOS VLSI Design: A Circuits and Systems Perspective*. Pearson Education, Inc., third edition, 2005. ISBN 0321149017.

20. Wikipedia. "Linear Feedback Shift Register", Nov 2007.

21. Wollinger, Thomas and Christof Paar. *New Algorithms, Architecture, and Applications for Reconfigurable Computing*, chapter Security Aspects of FPGAs in Cryptographic Applications. Kluwer, 2004.

22. Xilinx, Inc. *Virtex 2 User Guide*, v4.2 edition, November 5 2007.

23. Xilinx, Inc. *Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet*, v4.6 edition, March 5 2007.

Vita

James Crouch graduated from North Thurston High School in Lacey, WA in 1998. After high school, James enrolled at Washington State University in Pullman, WA and graduated with a Bachelors of Engineering in Electrical Engineering in 2003. James was commissioned into the USAF after graduation in 2003 and served first at Wright-Patterson AFB working for Sensors Directorate of AFRL. James then was assigned to AFIT and completed his masters in electrical engineering in 2008. After AFIT, James was stationed at Los Angeles AFB, working for the MDA.

Permanent address: 2950 Hobson Way
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
27-03-2008	**Master's Thesis**	Aug 06 – Mar 08

4. TITLE AND SUBTITLE
Digital Fingerprinting of Field Programmable Gate Arrays

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Crouch, James W., Captain, USAF

5d. PROJECT NUMBER
JON: 07-152

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7764

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/08-06

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Dr. Robert Bennington
AFRL/RYT
2241 Avionics Circle
WPAFB OH 45433
(937)320-9068 x111
Email: Robert.Bennington@wpafb.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Commercial off-the-shelf (COTS) component usage is becoming more prevalent in military applications due to current Department of Defense (DoD) policies. The easy accessibility of COTS will give reverse engineers a higher probability of successfully tampering, coping, or reverse engineering circuits that contain critical capabilities. To prevent this and verify the trustworthiness of hardware, circuit identification tags or serials numbers can be used. However, these values can be easily obtained and forged. To protect critical DoD technologies from possible exploitation, there is an urgent need for a reliable method to confirm a circuit's identity using a set of unique unforgettable metrics. This research proposes the concept of creating a circuit identifier, or digital fingerprint, for application specific integrated circuits (ASIC) and field programmable gate arrays (FPGAs). The digital fingerprint would be a function of the natural variations in the semiconductor manufacturing process and the functionality of the circuit allowing the creation of a unique identifier for a specific chip that can not be duplicated or forged. The proposed digital fingerprint allows the use of any arbitrary node or set of nodes internal to the circuit and the circuit outputs as monitoring locations. Changes in the signal on a selected node or output can be quantified digitally over a period of time or at a specific instance of time. Two monitoring methods are proposed, one using cumulative observation of the nodes and the other samples the nodes based on a signal transition. Testing of the two monitoring methods was performed on a small sample of twenty Xilinx® Virtex-II Pro FPGAs. Both methods successfully created unique identifiers for each FPGA.

15. SUBJECT TERMS
Digital fingerprint, unique identifier, field programmable gate array, application specific integrated circuit, asynchronous sampling, cumulative sampling, signal transitions, fabrication variations

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
REPORT	**ABSTRACT**	**c. THIS PAGE**			**Dr. Yong C. Kim (ENG)**
U	**U**	**U**	**UU**	87	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636 x4620 Email:yong.kim@afit.edu

Standard Form 298 (Rev: 8-98)
Prescribed by ANSI Std. Z39-18